تحليل المشاعر

(التنقيب في الآراء)

عن طريق الامثلة

٢٥ مشروع تحليل مشاعر تم حلها وشرحها باستخدام بايثون

ترجمة واعداد: د. علاء طعيمة



بسمه تعالى

تحليل المشاعر: عن طريق الامثلة

25 مشروع تحليل مشاعر تم حلها وشرحها باستخدام بايثون

ترجمة واعداد:

د. علاء طعيمة

المقدمة

تحليل المشاعر Sentiment analysis (المعروف أيضًا باسم التنقيب في الرأي opinion mining) هو نهج معالجة اللغة الطبيعية (NLP) لتحديد الإيجابية positivity أو السلبية negativity أو الحيادية neutrality البيانات. يتم استخدام تحليل المشاعر بشكل متكرر على البيانات النصية لمساعدة المؤسسات في تتبع مشاعر العلامة التجارية والمنتج في ملاحظات المستهلك وفهم متطلبات العملاء بشكل أفضل.

هل تبحث عن أفضل مشروع تحليل مشاعر؟ هل ترغب في اختبار معرفتك بتحليل المشاعر والتعلم الآلي؟ لقد جئت إلى المكان المناسب.في هذا الكتاب، سنمر بالعديد من مشاريع تحليل المشاعر التي قد تستخدمها لإثبات فهمك واختبار معلوماتك.

لقد حاولت قدر المستطاع ان اترجم المشاريع الأكثر طرحاً في مجال تحليل المشاعر مع الشرح المناسب والكافي، ومع هذا يبقى عملاً بشرياً يحتمل النقص، فاذا كان لديك أي ملاحظات حول هذا الكتاب، فلا تتردد بمراسلتنا عبر بريدنا الالكتروني alaa.taima@qu.edu.iq .

نأمل ان يساعد هذا الكتاب كل من يريد ان يدخل في مجالات المعالجة اللغوية الطبيعية وتحليل المشاعر ومساعدة القارئ العربي على تعلم هذه المجالات. اسأل الله التوفيق في هذا العمل لأثراء المحتوى العربي الذي يفتقر أشد الافتقار إلى محتوى جيد ورصين في المعالجة اللغوية الطبيعية وتحليل المشاعر. ونرجو لك الاستمتاع مع الكتاب ولا تنسونا من صالح الدعاء.

د. علاء طعيمة

كلية علوم الحاسوب وتكنولوجيا المعلومات

جامعة القادسية

العراق

المحتويات

0) ما هو تحليل المشاعر؟ What is Sentiment Analysis

أصبح تحليل المشاعر Sentiment analysis جزءًا لا يتجزأ من تسويق المنتجات وتجربة المستخدم، حيث تلجأ الشركات والمستهلكون على حد سواء إلى الموارد عبر الإنترنت للحصول على تعليقات على المنتجات والخدمات.في هذه المقالة، سأشرح ما هو تحليل المشاعرفي التعلم الآلي.

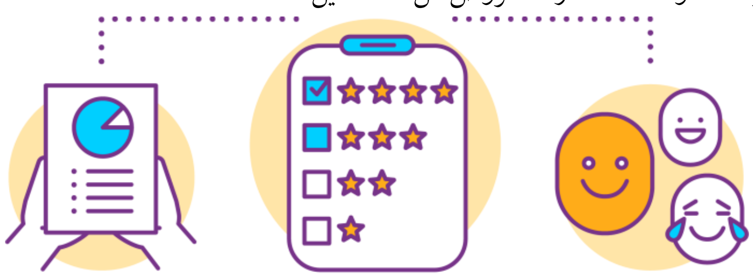
ما هو تحليل المشاعر؟

آراء الآخرين لها تأثير كبير على عملية صنع القرار اليومية لدينا. تتراوح هذه القرارات من شراء منتج مثل الهاتف الذكي إلى الاستثمارفي اختيار مدرسة، وجميع القرارات التي تؤثر على جوانب مختلفة من حياتنا اليومية. قبل الإنترنت، سعى الأشخاص للحصول على آراء حول المنتجات والخدمات من مصادر مثل الأصدقاء أو الأقارب أو تقارير المستهلكين.

ومع ذلك،في عصر الإنترنت، أصبح من الأسهل بكثير جمع الآراء المتنوعة من مختلف الأشخاصفي جميع أنحاء العالم. يتطلع الأشخاص إلى مراجعة المواقع ومواقع التجارة الإلكترونية ومواقع الرأي عبر الإنترنت ووسائل التواصل الاجتماعي للحصول على تعليقات حول كيفية عرض منتج أو خدمة معينةفي السوق. وبالمثل، تستخدم المنظمات استطلاعات الرأي polls واستطلاعات الرأي opinion polls ووسائل التواصل الاجتماعي كآلية للحصول على تعليقات على منتجاتها وخدماتها.

تحليل المشاعر هو الدراسة الحسابية للآراء والمشاعر والعواطف المعبر عنهافي النص.في علم البيانات والتعلم الآلي، يتم استخدام تحليل المشاعر بشكل متزايد لأن المعلومات التي يولدها يمكن أن تؤدي إلى تسييل المنتجات والخدمات.

على سبيل المثال، من خلال الحصول على تعليقات المستهلكين حول حملة تسويقية، يمكن للمؤسسة قياس نجاح الحملة أو تعلم تعديلها لتحقيق مزيد من النجاح. تعد مراجعات المنتجات Product reviews مفيدة أيضًا لإنشاء منتجات أفضل، والتي يمكن أن يكون لها تأثير مباشر على الإيرادات، وكذلك لمقارنة العروض من المنافسين.



Sentiment analysis

تعبر الكلمات عن أنواع مختلفة من المشاعر التي يمكن أن تكون إيجابية أو سلبية أو قوية أو ضعيفة. لإجراء تحليل المشاعر، من المهم فهم قطبية الكلمات وتصنيف المشاعر إلى فئات مثل الإيجابية positive أو السلبية negative أو المحايدة neutral. يمكن إنجاز هذه المهمة من خلال استخدام خوارزميات التعلم الآلي. باستخدام التعلم الآلي، يمكننا تصنيف ما إذا كانت الكلمات لها مشاعر إيجابية أو سلبية.

ومع ذلك، لا تزال هناك بعض التحديات التي يجب التغلب عليها قبل أن يصبح تحليل المشاعر أداة مثالية أكثر. على سبيل المثال، لا يزال الحكم البشري أكثر دقة كمقياس في تحليل المشاعر. لا تستطيع الأنظمة الآلية التمييز بين السخرية sarcasm والنص الصادق sincere text، ولا يمكنها دائمًا تحليل المعنى السياقي المحدد للكلمة بشكل صحيح. يطرح استخدام الاختصارات مثل "lol" أو اختصارات الكلمات مشاكل في التفسير.

بلا شك، هذا مجال متطور مع مجموعة متنوعة من التطبيقات المفيدة. على الرغم من صعوبة مهام تحليل المشاعر بسبب أصول معالجة اللغة الطبيعية، فقد كان هناك الكثير من التقدم في السنوات الأخيرة بسبب ارتفاع الطلب. لا ترغب الشركات فقط في معرفة كيف ينظر المستهلكون إلى منتجاتها وخدماتها، ولكن المستهلكين يريدون معرفة آراء الآخرين قبل اتخاذ قرارات الشراء.

مشاريع التعلم الآلي على تصنيف المشاعر

أتمنى أن تعرف الآن ما هو تحليل المشاعر وكيف يتم استخدام خوارزميات التعلم الآلي لمهمة تصنيف المشاعر. فيما يلي بعض مشاريع التعلم الآلي بناءً على تحليل المشاعر:

- التنبؤ بنتائج الانتخابات الرئاسية باستخدام التغريدات.
- نموذج كشف الكلام الذي يحض على الكراهية.
- تحليل مراجعات الفيلم.
- تحليل مراجعات منتجات أمازون.
- نموذج كشف الأخبار الكاذبة.

أتمنى أن تكون قد أحببت هذا المقال حول ماهية تحليل المشاعر وما فائدة التعلم الآلي فيه.

المصدر:

https://thecleverprogrammer.com/2020/12/24/what-is-sentiment-analysis/

1) تحليل المشاعر باستخدام بايثون Sentiment Analysis with Python

في التعلم الآلي Machine Learning، يشير تحليل المشاعر Sentiment analysis إلى تطبيق معالجة اللغة الطبيعية natural language processing واللغويات الحاسوبية computational linguistics وتحليل النص text analysis لتحديد وتصنيف الآراء الشخصيةفي المستندات المصدر.في هذه المقالة، سأقدم لك مشروع التعلم الآلي حول تحليل المشاعر باستخدام لغة برمجة بايثون.

ما هو تحليل المشاعر؟

يهدف تحليل المشاعر إلى تحديد موقف الكاتب تجاه موضوع أو القطبية السياقية العامة للمستند. يمكن أن يكون الموقف هو حكمه أو تقييمه أو حالته العاطفية أو التواصل العاطفي المقصود.

في تحليل المشاعر، تتمثل المهمة الرئيسيةفي تحديد كلمات الرأي opinion words، وهو أمر مهم للغاية. كلمات الرأي هي مؤشرات سائدة على المشاعر، وخاصة الصفات والظروف والأفعال، على سبيل المثال: "I love this camera. It’s amazing!"

تُعرف كلمات الرأي أيضًا باسم كلمات القطبية polarity words، أو كلمات المشاعر، أو معجم الرأي sentiment words، أو كلمات الرأي opinion words، والتي يمكن عمومًا تقسيمها إلى نوعين: الكلمات الإيجابية positive words، على سبيل المثال، رائع wonderful، أنيق elegant، مذهل astonishing. والكلمات السلبية negative words، على سبيل المثال فظيع horrible، مقرف disgusting، فقير poor.

مشروع التعلم الآلي حول تحليل المشاعر باستخدام بايثون

الآنفي هذا القسم، سوف آخذك خلال مشروع التعلم الآلي حول تحليل المشاعر باستخدام لغة برمجة بايثون. لنبدأ باستيراد جميع مكتبات بايثون ومجموعة البيانات dataset الضرورية:

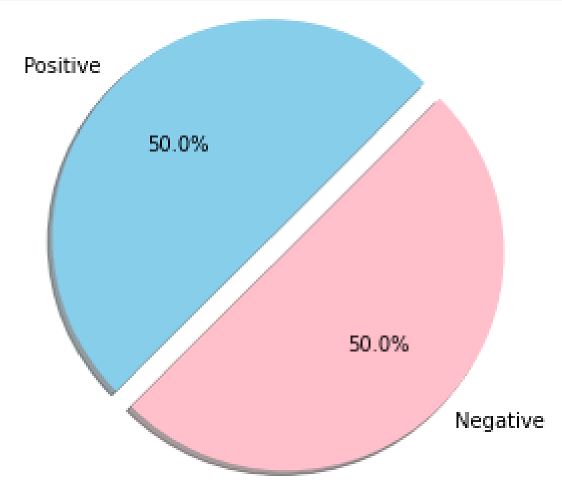
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
count=CountVectorizer()
data=pd.read_csv("Train.csv")
data.head()
```

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0
3	Even though I have great interest in Biblical ...	0
4	Im a die hard Dads Army fan and nothing will e...	1

بعد قراءة مجموعة البيانات التي تحتوي على 40 ألفًا من تقييمات الأفلام من IMDB، نرى أن هناك عمودين بارزين. أحدهما هو TEXT الذي يحتوي على النقد والآخر هو LABEL الذي يحتوي على O و 1، حيث 0 ـ سلبي و 1 ـ إيجابي.

الآن دعنا نرسم توزيع البيانات:

```
fig=plt.figure(figsize=(5,5))
colors=["skyblue",'pink']
pos=data[data['label']==1]
neg=data[data['label']==0]
ck=[pos['label'].count(),neg['label'].count()]
legpie=plt.pie(ck,labels=["Positive","Negative"],
                 autopct ='%1.1f%%',
                 shadow = True,
                 colors = colors,
                 startangle = 45,
                 explode=(0, 0.1))
```



ثم سنقوم باستيراد RE، أي عملية التعبير المنتظم regular expression، نستخدم هذه المكتبة لإزالة علامات html مثل "<a>" أو. لذلك كلما صادفنا هذه العلامات، فإننا نستبدلها بسلسلة

فارغة. ثم سنقوم أيضًا بتعديل الرموز التعبيرية التي يمكن أن تكون وجوهًا ضاحكة (:)، وجه حزين: (أو حتى وجه مستاء: /. سنقوم بتغيير الرموز التعبيريةفي النهاية للحصول على مجموعة نصية نظيفة:

```
import re
def preprocessor(text):
             text=re.sub('<[^>]*>','',text)
             emojis=re.findall('(?::|;|=)(?:-
)?(?:\)|\(|D|P)',text)
             text=re.sub('[\W]+',' ',text.lower()) +\
                ' '.join(emojis).replace('-','')
             return text
data['text']=data['text'].apply(preprocessor)
```

الآن، سأستخدم PorterStemmer من nltk لتبسيط البيانات وإزالة التعقيدات غير الضرورية في بياناتنا النصية:

```
from nltk.stem.porter import PorterStemmer
porter=PorterStemmer()
def tokenizer(text):
        return text.split()
def tokenizer_porter(text):
    return [porter.stem(word) for word in text.split()]
```

تصوير الكلمات السلبية والإيجابية

لتصوير الكلمات السلبية والإيجابية باستخدام سحابة الكلمات wordcloud، سأقوم أولاً بإزالة كلمات التوقف stopwords:

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop=stopwords.words('english')

from wordcloud import WordCloud
positivedata = data[ data['label'] == 1]
positivedata =positivedata['text']
negdata = data[data['label'] == 0]
negdata= negdata['text']

def wordcloud_draw(data, color = 'white'):
    words = ' '.join(data)
    cleaned_word = " ".join([word for word in words.split()
                              if(word!='movie' and
word!='film')
                            ])
    wordcloud = WordCloud(stopwords=stop,
                      background_color=color,
                      width=2500,
                      height=2000
                     ).generate(cleaned_word)
    plt.figure(1,figsize=(10, 7))
    plt.imshow(wordcloud)
    plt.axis('off')
```

```
    plt.show()

print("Positive words are as follows")
wordcloud_draw(positivedata,'white')
print("Negative words are as follows")
wordcloud_draw(negdata)
```



الكلمات الإيجابية التي يتم إبرازها هي love، excellent، perfect، good، beautiful، kind، والكلمات السلبية التي يتم إبرازها هي: horrible، wasteful، problem، stupid، horrible، bad، poor.

سأستخدم الآن TF-IDF Vertorizer لتحويل المستندات الأولية إلى مصفوفة ميزات feature matrix وهو أمر مهم جداً لتدريب نموذج التعلم الآلي:

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf=TfidfVectorizer(strip_accents=None,lowercase=False,prepr
ocessor=None,tokenizer=tokenizer_porter,use_idf=True,norm='l2'
,smooth_idf=True)
y=data.label.values
x=tfidf.fit_transform(data.text)
```

تدريب نموذج التعلم الآلي لتحليل المشاعر

الآن لتدريب نموذج التعلم الآلي، سأقسم البيانات إلى مجموعات تدريب بنسبة 50 بالمائة واختبار بنسبة 50 بالمائة:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,random_stat
e=1,test_size=0.5,shuffle=False)
```

دعنا الآن ندرب نموذج التعلم الآلي لمهمة تحليل المشاعر باستخدام نموذج الانحدار اللوجستي :Logistic Regression

```
from sklearn.linear_model import LogisticRegressionCV
clf=LogisticRegressionCV(cv=6,scoring='accuracy',random_state=
0,n_jobs=-1,verbose=3,max_iter=500).fit(X_train,y_train)
y_pred = clf.predict(X_test)
from sklearn import metrics

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.89045
```

أتمنى أن تكون قد أحببت هذه المقالة حول تحليل المشاعر باستخدام لغة برمجة بايثون.

المصدر:

https://thecleverprogrammer.com/2020/12/07/sentiment-analysis-with-python/

2) تحليل المشاعر مع التعلم الآلي Sentiment Analysis with machine learning

تحليل المشاعر Sentiment analysis هو العملية التي يمكن من خلالها قياس كل المحتوى كمياً لتمثيل أفكار ومعتقدات وآراء قطاعات كاملة من الجمهور. من الصعب التقليل من الآثار المترتبة على تحليل المشاعر لزيادة إنتاجية العمل. يعد تحليل المشاعر إحدى مهام البرمجة اللغوية العصبية NLP الشائعة التي يحتاج كل عالم بيانات لأدائها.

على سبيل المثال، أنت طالب في دورة تدريبية عبر الإنترنت ولديك مشكلة. قمت بنشره في منتدى الفصل. لن يكون تحليل المشاعر قادرًا على تحديد الموضوع الذي تعاني منه فحسب، بل أيضًا تحديد مدى إحباطك، وتخصيص تعليقاتهم وفقًا لهذا الشعور. هذا يحدث بالفعل لأن التكنولوجيا موجودة بالفعل.

تحليل المشاعر مع التعلم الآلي

آمل أن تكون قد فهمت ما يعنيه تحليل المشاعر. سأقدم لك الآن طريقة سهلة للغاية لتحليل المشاعر باستخدام التعلم الآلي. تتضمن البيانات التي سأستخدمها 27481 تغريدة تم وضع علامة عليهافي مجموعة التدريب و3534 تغريدةفي مجموعة الاختبار. يمكنك بسهولة تنزيل البيانات من هنا. لنبدأ الآن بهذه المهمة من خلال النظر إلى البيانات باستخدام Pandas:

```
import pandas as pd
training = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
print("Training data: \n",training.head())
print("Test Data: \n",test.head())
```

```
Training data:
        textID  ... sentiment
0  cb774db0d1  ...   neutral
1  549e992a42  ...  negative
2  088c60f138  ...  negative
3  9642c003ef  ...  negative
4  358bd9e861  ...  negative

[5 rows x 4 columns]
Test Data:
        textID                                               text sentiment
0  f87dea47db  Last session of the day  http://twitpic.com/67ezh   neutral
1  96d74cb729   Shanghai is also really exciting (precisely -...  positive
2  eee518ae67  Recession hit Veronique Branquinho, she has to...  negative
3  01082688c6                                        happy bday!  positive
4  33987a8ee5             http://twitpic.com/4w75p - I like it!!  positive
```

معالجة البيانات

من أجل البساطة، لا نريد المبالغةفي التعامل مع جانب تنظيف البيانات data cleaning، ولكن هناك بعض الأشياء البسيطة التي يمكننا القيام بها لمساعدة نموذج التعلم الآلي الخاص بنا على تحديد المشاعر. عملية تنظيف البيانات كالتالي:

1. إزالة جميع الارتباطات التشعبية (hyperlinks) من التغريدات.
2. استبدال الاختصارات (contractions) الشائعة.
3. إزالة علامات الترقيم (punctuation).

كعملية لإعداد البيانات، يمكننا إنشاء دالة لتعيين تسميات المشاعر إلى الأعداد الصحيحة وإعادتها من الدالة:

```
import re

contractions_dict = {"can`t": "can not,"
"                      won`t": "will not,"
"                      don`t": "do not,"
"                      aren`t": "are not,"
"                      i`d": "i would,"
"                      couldn`t": "could not,"
"                      shouldn`t": "should not,"
"                      wouldn`t": "would not,"
"                      isn`t": "is not,"
"                      it`s": "it is,"
"                      didn`t": "did not,"
"                      weren`t": "were not,"
"                      mustn`t": "must not,"
{


def prepare_data(df:pd.DataFrame) -> pd.DataFrame:

    df["text"] = df["text"]\
.               apply(lambda x: re.split('http:\/\/.*',
str(x))[0])\
.               str.lower\ ()
.               apply(lambda x:
replace_words(x,contractions_dict))

    df["label"] = df["sentiment"].map)
"}                          neutral": 1, "negative":0,
"positive":2{
(
    return df.text.values, df.label.values
def replace_words(string:str, dictionary:dict):
    for k, v in dictionary.items:()
        string = string.replace(k, v)
    return string
train_tweets, train_labels = prepare_data(train_df)
```

```
test_tweets, test_labels = prepare_data(test_df)
```

الترميز Tokenization

نحتاج الآن إلى تحويل كل تغريدة إلى رمز متجه واحد ذي طول ثابت ـ على وجه التحديد تكامل TFIDF. للقيام بذلك، يمكننا استخدام Tokenizer() المدمج في Keras ، وهو مناسب لبيانات التدريب:

```
from keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_tweets)
train_tokenized =
tokenizer.texts_to_matrix(train_tweets,mode='tfidf')
test_tokenized =
tokenizer.texts_to_matrix(test_tweets,mode='tfidf')
```

نموذج التعلم الآلي لتحليل المشاعر

الآن، سأقوم بتدريب نموذجنا على تحليل المشاعر باستخدام خوارزمية تصنيف الغابات العشوائية Random Forest Classification المقدمة من Scikit-Learn:

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=500,
min_samples_leaf=2,oob_score=True,n_jobs=-1,)
forest.fit(train_tokenized,train_labels)
print(f"Train score:
{forest.score(train_tokenized,train_labels)}")
print(f"OOB score: {forest.oob_score_}")
```

Train score: 0.7672573778246788
OOB score: 0.6842545758887959

تقييم النموذج في مجموعة الاختبار

تسهل Scikit-Learn استخدام كل من المصنف وبيانات الاختبار لإنتاج خوارزمية مصفوفة الارتباك confusion matrix تظهر الأداء في مجموعة الاختبار على النحو التالي:

```
print("Test score: ",forest.score(test_tokenized,test_labels))
```

Test score: 0.687889077532541

معدل الدقة accuracy rate ليس كبيرًا لأن معظم أخطائنا تحدث عند توقع الفرق بين المشاعر الإيجابية والسلبية والحيادية، والتي في المخطط الكبير للأخطاء ليست أسوأ شيء يمكن أن نمتلكه. على الرغم من أننا لحسن الحظ، نادرًا ما نخلط بين الشعور الإيجابي والشعور السلبي والعكس صحيح.

أتمنى أن تكون قد أحببت هذا المقال عن تحليل المشاعر.

المصدر:

/https://thecleverprogrammer.com/2020/08/13/sentiment-analysis

3) تحليل مشاعر مراجعات Tinder باستخدام بايثون Tinder Reviews Sentiment Analysis using Python

Tinder هو أحد أكثر تطبيقات (dating) المواعدة شيوعًا. يربط الأشخاص الذين لديهم اهتمامات مماثلة. للتحقق مما إذا كان Tinder يساعد الأشخاص في العثور على شركاء، يمكننا تحليل مشاعر الناس حول Tinder. هناك الكثير من المراجعات على متجر Google Play حول Tinder. يمكننا استخدام هذه البيانات لتحليل مشاعر مستخدمي Tinder. لذلك إذا كنت تريد معرفة كيفية تحليل مراجعات Tinder، فهذه المقالة مناسبة لك. في هذه المقالة، سوف آخذك خلال مهمة مراجعة Tinder لتحليل المشاعر باستخدام بايثون.

تحليل مشاعر مراجعات Tinder باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل آراء Tinder من Kaggle. تم جمعها من مراجعات Tinder على متجر Google Play. دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة لبدء هذه المهمة:

```
import pandas as pd
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
import nltk
import re
from nltk.corpus import stopwords
import string
data = pd.read_csv("tinder_google_play_reviews.csv")
print(data.head())
```

```
                                             reviewId      userName  \
0  gp:AOqpTOF5m-nY12XsKXO0IG-ZQtyvmjwKEp43ILLrhBS...    Kreg Smith
1  gp:AOqpTOFMaTJ6Mj-6hrp6ZI9gU5fzeVZQA9LugbFe1xR...          R.W.
2  gp:AOqpTOGtOLC4xZzU1NT8t1ykvQHfOuhW7oJ0MScukLj...  Benjo cantor
3  gp:AOqpTOGcid22sko0XyvhV1kSpbdKUzx5Q1SIi5L1Ovc...   Chris Plata
4  gp:AOqpTOGzA20eNWEOUM8edTHGQfd6OU7Qy48JpUcBT-x...    Dave Midas

                                           userImage  \
0  https://play-lh.googleusercontent.com/a/AATXAJ...
1  https://play-lh.googleusercontent.com/a-/AOh14...
2  https://play-lh.googleusercontent.com/a/AATXAJ...
3  https://play-lh.googleusercontent.com/a/AATXAJ...
4  https://play-lh.googleusercontent.com/a-/AOh14...

                                             content  score  thumbsUpCount  \
0  Got banned for life don't know why they won't ...      1              0
1  I don't know why I was banned .. But I m not a...      1              0
2                        All gays even if your straight🧑🧑🧑      1              0
3  You have to pay so much to even be seen on thi...      1              0
4  I do not understand how so many people use thi...      2              0
```

```
  reviewCreatedVersion                  at replyContent repliedAt
0              13.6.1  2022-05-21 04:10:44          NaN       NaN
1                 NaN  2022-05-21 04:08:24          NaN       NaN
2              13.6.1  2022-05-21 04:00:10          NaN       NaN
3                 NaN  2022-05-21 03:47:58          NaN       NaN
4              13.6.1  2022-05-21 03:47:51          NaN       NaN
```

في الانطباعات الأولى لمجموعة البيانات هذه، يمكنني رؤية بعض القيم الخاليةفي بعض الأعمدة. لتحليل مراجعات Tinder، نحتاج فقط إلى عمود المحتوى (content column). لذلك دعونا ننشئ مجموعة بيانات جديدة مع عمود المحتوى ونتحرك إلى أبعد من ذلك:

```
data = data[["content"]]
```

دعنا الآن نرى ما إذا كانت لدينا قيم خاليةفي عمود المحتوى:

```
data.isnull().sum()
```

يحتوي عمود المحتوى أيضًا على قيم خالية، فلنقم بإزالة القيم الخالية والمضي قدمًا:

```
data = data.dropna()
```

دعنا الآن نجهز هذه البيانات لمهمة تحليل المشاعر (sentiment analysis). هنا يتعين علينا تنظيف النص في عمود المحتوى:

```
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
stopword=set(stopwords.words('english'))

def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["content"] = data["content"].apply(clean)
```

دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الأشخاص في مراجعات Tinder:

```
text = " ".join(i for i in data.content)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
```

```
plt.show()
```

سأضيف الآن ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها ايجابية " Positive" وسلبية "Negative" ومحايدة "Neutral" من خلال حساب درجات المشاعر (sentiment scores) للمراجعات:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["content"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["content"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["content"]]
data = data[["content", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                             content  Positive  Negative  \
0  got ban life dont know wont tell help way read...     0.231     0.225
1  dont know ban  allow use tinder henceforth did...     0.371     0.108
2                              gay even straight 👱‍♀️👱‍♀️👱‍♀️     0.000     0.000
3  pay much even seen app girl ever talk back bet...     0.229     0.128
4  understand mani peopl use app im mean ugli ive...     0.000     0.225

   Neutral
0    0.544
1    0.521
2    1.000
3    0.642
4    0.775
```

دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الأشخاص في التعليقات الإيجابية على Tinder:

```
positive =' '.join([i for i in
data['content'][data['Positive'] > data["Negative"]]])
```

```
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(positive)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الأشخاص في المراجعات السلبية لـ Tinder:

```
negative =' '.join([i for i in
data['content'][data['Negative'] > data["Positive"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(negative)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



دعنا نلقي نظرة على النتيجة الإجمالية لمشاعر مستخدمي Tinder :

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
  if (a>b) and (a>c):
    print("Positive 😊 ")
  elif (b>a) and (b>c):
    print("Negative 😠 ")
  else:
    print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك يكتب معظم المستخدمين تعليقات محايدة. دعونا نلقي نظرة على إجمالي جميع درجات المشاعر:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  158277.42200002735
Negative:  59438.14199999961
Neutral:  314250.34899997106
```

كما ترى، الإيجابي أكثر بكثير من السلبي، يمكننا القول إن معظم المستخدمين سعداء بـ Tinder.

الملخص

هذه هي الطريقة التي يمكنك بها تنفيذ مهمة Tinder تحليل مشاعر مراجعات Tinder باستخدام بايثون. هو أحد أكثر تطبيقات المواعدة شيوعًا. يربط الأشخاص الذين لديهم اهتمامات مماثلة. آمل أن تكون قد أحببت هذا المقال حول تحليل المراجعات في Tinder.

المصدر:

https://thecleverprogrammer.com/2022/05/24/tinder-reviews-sentiment-analysis-using-python/

4) تحليل المشاعر لمراجعات TikTok باستخدام بايثون TikTok Reviews Sentiment Analysis using Python

يعد TikTok أحد أشهر تطبيقات الوسائط الاجتماعية اليوم. تشتهر بمقاطع الفيديو القصيرة الخاصة بها. غالبًا ما يستخدم الأشخاص هذا التطبيق لمشاهدة مقاطع فيديو مسلية ومضحكة. على الرغم من محتواه الترفيهي، إلا أن هذا التطبيق لا يحبه الجميع. إذا تصفحت مراجعاته، فستجد مزيجًا من الكراهية والدعم لـ TikTokفي جميع أنحاء العالم. لذلك، دعنا نحلل مراجعات TikTok لمعرفة ما يشعر به الناس حول محتوى هذا التطبيق.في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعر لمراجعات TikTok باستخدام بايثون.

تحليل المشاعر لمراجعات TikTok باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها هنا لمهمة مراجعات TikTok لتحليل المشاعر من Kaggle. تم جمعهافي الأصل من تقييمات TikTok على متجر Google Play. يمكنك تنزيل مجموعة البيانات هذه من هنا. دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة لبدء هذه المهمة:

```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.corpus import stopwords
import string
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")

data = pd.read_csv("tiktok.csv")
print(data.head())
```

```
                                          content  score  thumbsUpCount  \
0                                            Good      5              0
1  Awesome app! Too many people on it where it's ...      5              0
2                                         Not bad      5              0
3                                        It is good      2              0
4                              Very interesting app      5              0

  reviewCreatedVersion                   at replyContent repliedAt
0               23.8.4  2022-04-05 23:18:30          NaN       NaN
1                  NaN  2022-04-05 23:18:21          NaN       NaN
2               23.9.5  2022-04-05 23:17:34          NaN       NaN
3               22.2.5  2022-04-05 23:17:04          NaN       NaN
4               22.1.5  2022-04-05 23:17:04          NaN       NaN
```

في الانطباعات (impressions) الأولى لمجموعة البيانات هذه، يمكنني رؤية قيم خاليةفي بعض الأعمدة. لتحليل تقييمات TikTok، نحتاج فقط إلى عمودين، المحتوى (content) والنتيجة

(score)؛ لذلك دعونا ننشئ مجموعة بيانات جديدة مع هذين العمودين فقط ونتقدم خطوة أخرى من خلال تحليل آراء TikTok:

```
data = data[["content", "score"]]
print(data.head())
```

```
                                         content  score
0                                           Good      5
1  Awesome app! Too many people on it where it's ...      5
2                                        Not bad      5
3                                     It is good      2
4                           Very interesting app      5
```

دعنا الآن نرى ما إذا كان أي من هذين العمودين يحتوي على أي قيم فارغة:

```
print(data.isnull().sum())
```

```
content    4
score      0
dtype: int64
```

لذلك هناك أربع قيم فارغةفي عمود المحتوى. دعنا نسقط القيم الفارغة ونتحرك أبعد من ذلك:

```
data = data.dropna()
```

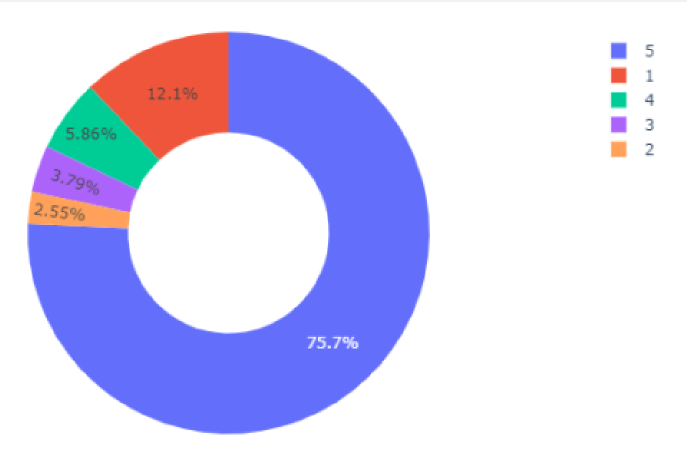
دعنا الآن نجهز هذه البيانات لمهمة تحليل المشاعر. هنا يتعين علينا تنظيف النصفي عمود المحتوى:

```
stopword=set(stopwords.words('english'))
def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["content"] = data["content"].apply(clean)
```

دعنا الآن نلقي نظرة على النسب المئوية للتقييمات الممنوحة لـ TikTok على متجر Google Play:

```
ratings = data["score"].value_counts()
```

```
numbers = ratings.index
quantity = ratings.values
import plotly.express as px
figure = px.pie(data,
      values=quantity,
      names=numbers,hole = 0.5)
figure.show()
```



يمكنك أن ترى أن 75.7٪ من المستخدمين قد أعطوا خمسة تقييمات لـ TikTok ، و 12.1٪ من المستخدمين صنفوها بـ 1. الآن دعنا نلقي نظرة على نوع الكلمات التي يستخدمها المستخدمونفي مراجعات TikTok:

```
text = " ".join(i for i in data.content)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



سأضيف الآن ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) للتغريدات:

دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الناس في التعليقات الإيجابية على TikTok:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["content"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["content"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["content"]]
data = data[["content", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                         content  Positive  Negative  \
0                                           good     1.000       0.0
1  awesom app mani peopl easier fb girl awesom gu...     0.381       0.0
2                                            bad     0.000       1.0
3                                           good     1.000       0.0
4                                    interest app     0.750       0.0

   Neutral
0    0.000
1    0.619
2    0.000
3    0.000
4    0.250
```

دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الناس غالبًا في التعليقات الإيجابية على TikTok:

```
positive =' '.join([i for i in
data['content'][data['Positive'] > data["Negative"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(positive)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الناس غالبًافي التعليقات السلبية على TikTok:

```
negative =' '.join([i for i in
data['content'][data['Negative'] > data["Positive"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(negative)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات TikTok باستخدام لغة برمجة بايثون.

الملخص

إذن هذه هي كيفية تحليل مشاعر مراجعات TikTok . تشتهر TikTok بمقاطع الفيديو القصيرة الخاصة بها. غالبًا ما يستخدم الأشخاص هذا التطبيق لمشاهدة مقاطع فيديو مسلية

ومضحكة. على الرغم من محتواه الترفيهي، إلا أن هذا التطبيق لا يحبه الجميع. آمل أن تكون قد أحببت هذه المقالة حول تحليل مشاعر مراجعات TikTok باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2022/04/12/tiktok-reviews-sentiment-analysis-using-python/

5) تحليل مشاعر حرب أوكرانيا وروسيا على تويتر باستخدام بايثون
Ukraine Russia War Twitter Sentiment Analysis using Python

اليوم هو اليوم التاسع عشر للحرب بين روسيا وأوكرانيا. تدعم العديد من الدول أوكرانيا من خلال فرض عقوبات اقتصادية على روسيا. هناك الكثير من التغريدات حول حرب أوكرانيا وروسيا حيث يميل الناس إلى تحديث الحقائق على الأرض، وما يشعرون به حيال ذلك، ومن يدعمون. لذلك إذا كنت ترغب في تحليل مشاعر الناس بشأن الحرب في أوكرانيا وروسيا، فهذا المقال مناسب لك. في هذه المقالة، سأطلعك على مهمة تحليل مشاعر الحرب في أوكرانيا وروسيا على تويتر باستخدام لغة بايثون.

تحليل معنويات حرب أوكرانيا وروسيا على تويتر باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل المشاعر على Twitter بشأن حرب أوكرانيا وروسيا من Kaggle. تم جمع مجموعة البيانات هذه في البداية من Twitter ويتم تحديثها بانتظام. يمكنك تنزيل مجموعة البيانات هذه من هنا. دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة للبدء بهذه المهمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
import nltk
import re
from nltk.corpus import stopwords
import string

data = pd.read_csv)
```

```
             id  conversation_id               created_at       date     time  \
0  1.502530e+18     1.502260e+18  2022-03-12 06:03:14 UTC  3/12/2022  6:03:14
1  1.502530e+18     1.502530e+18  2022-03-12 06:03:14 UTC  3/12/2022  6:03:14
2  1.502530e+18     1.502530e+18  2022-03-12 06:03:13 UTC  3/12/2022  6:03:13
3  1.502530e+18     1.502210e+18  2022-03-12 06:03:12 UTC  3/12/2022  6:03:12
4  1.502530e+18     1.500440e+18  2022-03-12 06:03:12 UTC  3/12/2022  6:03:12

   timezone       user_id         username  \
0         0  2.019880e+07         redcelia
1         0  2.275356e+08          eee_eff
2         0  8.431317e+07      mistify_007
3         0  9.898620e+17  reallivinghuman
4         0  1.164940e+18           rpcsas
```

```
                                   name place  ... geo source user_rt_id  \
0    Johnson OutuA EUITAF💙😷 #NeverVoteTory    NaN  ... NaN    NaN        NaN
1  Wearing Masks still saves lives UAMC 🌹 🌹     NaN  ... NaN    NaN        NaN
2                              Brian        NaN  ... NaN    NaN        NaN
3                                   Basha    NaN  ... NaN    NaN        NaN
4                                  RonJon    NaN  ... NaN    NaN        NaN

  user_rt retweet_id                                           reply_to  \
0     NaN        NaN  [{'screen_name': 'RussianEmbassy', 'name': 'Ru...
1     NaN        NaN                                                 []
2     NaN        NaN                                                 []
3     NaN        NaN  [{'screen_name': 'RussianEmbassy', 'name': 'Ru...
4     NaN        NaN  [{'screen_name': 'IsraeliPM', 'name': 'Prime M...

   retweet_date  translate trans_src trans_dest
0           NaN        NaN       NaN        NaN
1           NaN        NaN       NaN        NaN
2           NaN        NaN       NaN        NaN
3           NaN        NaN       NaN        NaN
4           NaN        NaN       NaN        NaN
```

دعونا نلقي نظرة سريعة على جميع أسماء الأعمدة لمجموعة البيانات:

```
print(data.columns)
```

```
Index(['id', 'conversation_id', 'created_at', 'date', 'time', 'timezone',
       'user_id', 'username', 'name', 'place', 'tweet', 'language', 'mentions',
       'urls', 'photos', 'replies_count', 'retweets_count', 'likes_count',
       'hashtags', 'cashtags', 'link', 'retweet', 'quote_url', 'video',
       'thumbnail', 'near', 'geo', 'source', 'user_rt_id', 'user_rt',
       'retweet_id', 'reply_to', 'retweet_date', 'translate', 'trans_src',
       'trans_dest'],
      dtype='object')
```

نحتاج فقط إلى ثلاثة أعمدة لهذه المهمة (اسم المستخدم (username) والتغريدة (tweet) واللغة (language))؛ سأختار فقط هذه الأعمدة وأمضي قدمًا:

```
data = data[["username", "tweet", "language"]]
```

دعنا نلقي نظرة على ما إذا كان أي من هذه الأعمدة يحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
username    0
tweet       0
language    0
dtype: int64
```

لذلك لا يحتوي أي من الأعمدة على قيم فارغة، فلنلقِ نظرة سريعة على عدد التغريدات التي يتم نشرها بأي لغة:

```
data["language"].value_counts()
```

```
en     8812
pt      251
und     198
it      155
in      122
ru       85
hi       55
ja       52
es       40
ta       23
tr       19
ca       18
fr       16
et       16
tl       15
nl       14
de       13
pl       13
fi        9
ar        9
zh        9
sv        6
uk        6
te        6
mr        5
cs        4
el        4
gu        4
no        3
th        3
kn        3
ro        3
ur        2
or        2
eu        2
ko        2
ht        2
sl        2
bn        1
cy        1
ne        1
Name: language, dtype: int64
```

لذا فإن معظم التغريدات باللغة الإنجليزية. دعنا نجهز هذه البيانات لمهمة تحليل المشاعر. سأقوم هنا بإزالة جميع الروابط وعلامات الترقيم والرموز وأخطاء اللغة الأخرى من التغريدات:

```
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
stopword=set(stopwords.words('english'))

def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
```

```
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["tweet"] = data["tweet"].apply(clean)
```

دعنا الآن نلقي نظرة على سحابة الكلمات (wordcloud) في التغريدات، والتي ستظهر الكلمات الأكثر استخدامًا في التغريدات من قبل الأشخاص الذين يشاركون مشاعرهم والتحديثات حول حرب أوكرانيا وروسيا:

```
text = " ".join(i for i in data.tweet)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



سأضيف الآن ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) للتغريدات:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["tweet"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["tweet"]]
```

```
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["tweet"]]
data = data[["tweet", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                               tweet  Positive  Negative  \
0  russianembassi ft mfarussia jeffdsach csdcolum...     0.077     0.284
1  kidnap without charg access lawyer putin russi...     0.000     0.000
2  much western civil everyon feel compel find cr...     0.144     0.259
3  russianembassi love place ill visit sure next ...     0.291     0.126
4  israelipm iaeaorg didnt know state israel advi...     0.000     0.000

   Neutral
0    0.639
1    1.000
2    0.596
3    0.583
4    1.000
```

دعنا الآن نلقي نظرة على الكلمات الأكثر شيوعًا التي يستخدمها الأشخاص ذوو المشاعر الإيجابية:

```
positive =' '.join([i for i in data['tweet'][data['Positive']
> data["Negative"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(positive)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



دعنا الآن نلقي نظرة على الكلمات الأكثر شيوعًا التي يستخدمها الأشخاص ذوو المشاعر السلبية:

```
negative =' '.join([i for i in data['tweet'][data['Negative']
> data["Positive"]]])
stopwords = set(STOPWORDS)
```

```
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(negative)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



هذه هي الطريقة التي يمكنك بها تحليل مشاعر الناس بشأن حرب أوكرانيا وروسيا. آمل أن تنتهي هذه الحرب قريبًا وأن تعود الأمور إلى طبيعتها.

الملخص

هناك الكثير من التغريدات حول حرب أوكرانيا وروسيا حيث يميل الناس إلى تحديث الحقائق على الأرض، وما يشعرون به حيال ذلك، ومن يدعمون. لقد استخدمت هذه التغريداتفي مهمة تحليل المشاعر على تويتر بشأن حرب أوكرانيا وروسيا.

المصدر:

https://thecleverprogrammer.com/2022/03/15/ukraine-russia-war-
/twitter-sentiment-analysis-using-python

6) تحليل مشاعر مراجعات Flipkart باستخدام بايثون Flipkart Reviews Sentiment Analysis using Python

Flipkart هي واحدة من أشهر الشركات الهندية. إنها منصة للتجارة الإلكترونية تتنافس مع منصات التجارة الإلكترونية الشهيرة مثل Amazon. واحدة من أكثر حالات استخدام علم البيانات شيوعًا هي مهمة تحليل المشاعر لمراجعات المنتجات المباعة على منصات التجارة الإلكترونية. لذلك، إذا كنت تريد معرفة كيفية تحليل مشاعر مراجعات Flipkart، فهذه المقالة مناسبة لك. في هذه المقالة، سوف أطلعك على مهمة Flipkart لمراجعة تحليل المشاعر باستخدام بايثون.

تحليل مشاعر مراجعات Flipkart باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها هنا لتحليل المشاعر لمراجعات Flipkart من Kaggle. لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/flipkart_reviews.csv")
print(data.head())
```

```
                                   Product_name  ... Rating
0  Lenovo Ideapad Gaming 3 Ryzen 5 Hexa Core 5600...  ...      5
1  Lenovo Ideapad Gaming 3 Ryzen 5 Hexa Core 5600...  ...      5
2  Lenovo Ideapad Gaming 3 Ryzen 5 Hexa Core 5600...  ...      5
3  DELL Inspiron Athlon Dual Core 3050U - (4 GB/2...  ...      5
4  DELL Inspiron Athlon Dual Core 3050U - (4 GB/2...  ...      5

[5 rows x 3 columns]
```

تحتوي مجموعة البيانات هذه على ثلاثة أعمدة فقط. دعنا نلقي نظرة على ما إذا كان أي من هذه الأعمدة يحتوي على قيم مفقودة أم لا:

```
print(data.isnull().sum())
```

```
Product_name    0
Review          0
Rating          0
dtype: int64
```

لذلك لا تحتوي مجموعة البيانات على أي قيم فارغة. نظرًا لأن هذه هي مهمة تحليل المشاعر لمراجعات Flipkart ، فسوف أقوم بتنظيف وإعداد العمود الذي يحتوي على المراجعات قبل التوجه إلى تحليل المشاعر:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

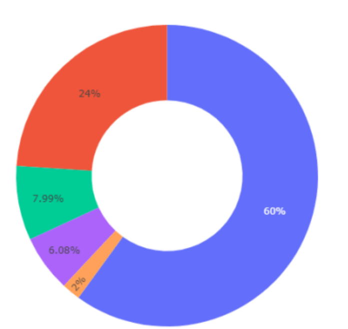
def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["Review"] = data["Review"].apply(clean)
```

تحليل المشاعر لمراجعاتFlipkart

يحتوي عمود التصنيف (Rating column) الخاص بالبيانات على التصنيفات التي قدمها كل مراجع. لذلك دعونا نلقي نظرة على كيفية قيام معظم الأشخاص بتقييم المنتجات التي يشترونها من Flipkart :

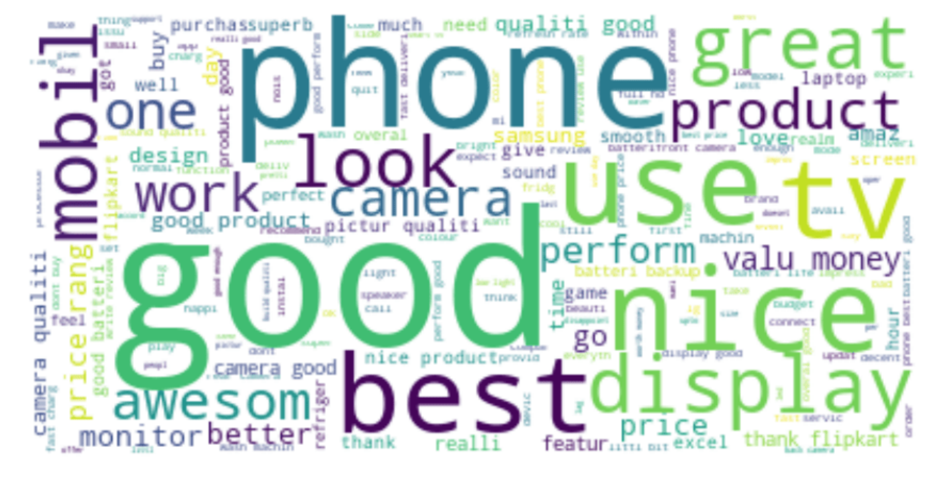
```
ratings = data["Rating"].value_counts()
numbers = ratings.index
quantity = ratings.values

import plotly.express as px
figure = px.pie(data,
     values=quantity,
     names=numbers,hole = 0.5)
figure.show()
```



لذلك منح 60 ٪ من المراجعين 5 من أصل 5 تقييمات للمنتجات التي يشترونها من Flipkart دعونا الآن نلقي نظرة على نوع التعليقات التي يتركها الأشخاص. لهذا، سأستخدم سحابة الكلمات (word cloud) لتصوير الكلمات الأكثر استخدامًافي عمود المراجعات :

```
text = " ".join(i for i in data.Review)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
          background_color="white").generate(text(
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



سأقوم الآن بتحليل مشاعر مراجعات Flipkart عن طريق إضافة ثلاثة أعمدةفي مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) للمراجعات:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Review"]]
```

```
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Review"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Review"]]
data = data[["Review", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                              Review  ...  Neutral
0  best  great performancei got around  backup bi...  ...    0.504
1                                        good perfom  ...    0.256
2  great perform usual also game laptop issu batt...  ...    0.723
3                     wife happi best product 👌🏻 😘  ...    0.488
4  light weight laptop new amaz featur batteri li...  ...    1.000

[5 rows x 4 columns]
```

الآن دعونا نرى كيف يفكر معظم المراجعين في منتجات وخدمات Flipkart:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
  if (a>b) and (a>c):
    print("Positive 😊 ")
  elif (b>a) and (b>c):
    print("Negative 😠 ")
  else:
    print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذا فإن معظم المراجعات محايدة. دعنا نلقي نظرة على إجمالي درجات المشاعر الإيجابية والسلبية والحيادية للعثور على نتيجة حول مراجعات Flipkart :

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  923.5529999999985
Negative:  96.77500000000013
Neutral:  1283.6880000000006
```

الملخص

لذلك، يعطي معظم الأشخاص مراجعات محايدة، وتقدم نسبة صغيرة من الأشخاص مراجعات سلبية. لذلك يمكننا القول إن الناس راضون عن منتجات وخدمات Flipkart. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعر لـ Flipkart باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2022/02/15/flipkart-reviews-/sentiment-analysis-using-python

7) تحليل المشاعر تجاه لقاح فايزر باستخدام بايثون Pfizer Vaccine Sentiment Analysis using Python

يعد Twitter أحد أكثر تطبيقات الوسائط الاجتماعية شيوعًا حيث يتمتع الأشخاص بحرية مشاركة آرائهم حول أي موضوع. هناك العديد من التغريدات المسجلة حول التوعية بلقاح فايزر (Pfizer vaccine) والتي يمكن استخدامها لتحليل مشاعر الناس حول لقاح فايزر. لذلك، إذا كنت تريد معرفة كيفية استخدام مجموعة بيانات Twitter لتحليل المشاعر، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعر تجاه لقاح فايزر باستخدام بايثون.

تحليل المشاعر تجاه لقاح فايزر باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل المشاعر تجاه لقاح فايزر من Kaggle، والتي تم جمعهافي البداية من Twitter عندما كان الأشخاص يشاركون آرائهم حول لقاح فايزر. لنبدأ مهمة تحليل المشاعر تجاه لقاح Pfizer عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/vaccination_tweets.csv")
print(data.head())
```

```
                    id            user_name  ... favorites is_retweet
0  1340539111971516416          Rachel Roh  ...         0      False
1  1338158543359250433         Albert Fong  ...         1      False
2  1337858199140118533           eliLTEU 👌  ...         0      False
3  1337855739918835717       Charles Adler  ...      2129      False
4  1337854064604966912  Citizen News Channel  ...         0      False

[5 rows x 16 columns]
```

لذا فإن مجموعة البيانات هذه كبيرة جدًا، دعنا نلقي نظرة على ما إذا كانت تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
id                     0
user_name              0
user_location       1630
user_description     506
user_created           0
user_followers         0
user_friends           0
user_favourites        0
user_verified          0
date                   0
text                   0
hashtags            1949
source                 1
retweets               0
favorites              0
is_retweet             0
dtype: int64
```

على الرغم من أن هذه القيم الخالية لن تؤثر على مهمة تحليل المشاعر، للحفاظ على بساطة الأمور، سأقوم بإسقاط الصفوف التي تحتوي على قيم خالية لأن مجموعة البيانات كبيرة بالفعل:

```
data = data.dropna()
```

```
                 id  user_followers  ...     retweets    favorites
count  4.749000e+03    4.749000e+03  ...  4749.000000  4749.000000
mean   1.355333e+18    5.069683e+04  ...     1.545378     9.385555
std    1.280104e+16    3.545440e+05  ...    13.395572    55.280915
min    1.337728e+18    0.000000e+00  ...     0.000000     0.000000
25%    1.344929e+18    1.740000e+02  ...     0.000000     0.000000
50%    1.352030e+18    6.480000e+02  ...     0.000000     1.000000
75%    1.364940e+18    2.728000e+03  ...     1.000000     5.000000
max    1.384788e+18    1.371493e+07  ...   678.000000  1979.000000

[8 rows x 6 columns]
```

يعد عمود النص (text column) أهم ميزةفي مجموعة البيانات هذه لأنه يحتوي على آراء مستخدمي Twitter حول لقاح فايزر .لكن يجب إعداد عمود النص لأنه يحتوي على العديد من الرموز الخاصة والأخطاء اللغوية. فيما يلي كيف يمكننا تنظيف عمود النص:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
```

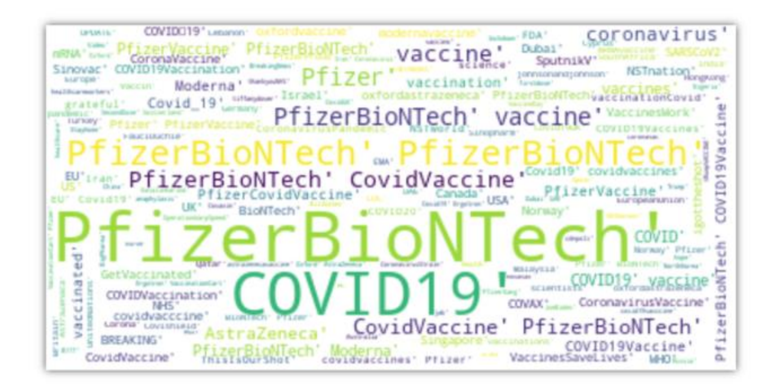
```
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["text"] = data["text"].apply(clean)
```

الآن، دعنا نلقي نظرة على سحابة الكلمات (word cloud)في عمود النص. سحابة الكلمات هي تقنية تصوير البيانات تعرض الكلمات الأكثر استخدامًا بخط كبير والكلمات الأقل استخدامًا بخط صغير. إليك كيفية تصوير سحابة الكلماتفي عمود النص:

```
text = " ".join(i for i in data.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

دعنا الآن نلقي نظرة على سحابة الكلماتفي عمود الوسوم (hashtags column)، والتي يمكن أن توضح نوع الوسوم التي كانت رائجة عندما كان الأشخاص يشاركون آرائهم حول لقاح فايزر:

```
text = " ".join(i for i in data.hashtags)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



يُظهر عمود (user_verified) في مجموعة البيانات ما إذا كان قد تم التحقق من المستخدمين الذين شاركوا آرائهم بواسطة Twitter أم لا. المستخدم الموثق على Twitter هو شخصية عامة أو شخصية مشهورة. لذلك دعونا نلقي نظرة على عدد المستخدمين الذين تم التحقق منهم والذين شاركوا آرائهم حول لقاح فايزر:

```
data["user_verified"].value_counts()
```

```
False    4169
True      580
Name: user_verified, dtype: int64
```

في الإخراج أعلاه، يُظهر False عدد المستخدمين الذين لم يتم التحقق منهم ويظهر True عدد المستخدمين الذين تم التحقق منهم. دعنا الآن ننتقل إلى مهمة تحليل المشاعر للقاح فايزر. سأضيف هنا ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر في عمود النص:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["text"]]
data = data[["text", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                           text  ...  Neutral
0   folk said daikon past could treat cytokin stor...  ...    0.748
2   coronavirus sputnikv astrazeneca pfizerbiontec...  ...    1.000
6   bit sad claim fame success vaccin patriot comp...  ...    0.481
9   covidvaccin state start get  monday us say pak...  ...    1.000
10  death close  mark million peopl wait pfizerbio...  ...    0.698

[5 rows x 4 columns]
```

الآن دعونا نحسب شعور معظم الناس تجاه لقاح فايزر:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك كانت معظم آراء المستخدمين محايدة، دعنا نلقي نظرة على إجمالي كل نتيجة عاطفية قبل التوصل إلى أي استنتاج:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  417.81600000000003
Negative:  188.81200000000024
Neutral:  4142.3750000000055
```

مجموع الإيجابيات والسلبيات أقل بكثير من الحيادية، لذلك يمكننا القول إن مناقشة مستخدمي تويتر كانت حول الوعي بلقاح فايزر بدلاً من مشاركة فوائده أو عيوبه.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل آراء مستخدمي تويتر حول لقاح فايزر.في ختام تحليل المشاعر هذا، يمكنني القول فقط إن مناقشة مستخدمي تويتر كانت حول الوعي بلقاح فايزر بدلاً من مشاركة فوائده أو عيوبه. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعر تجاه لقاح فايزر باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/10/12/pfizer-vaccine-/sentiment-analysis-using-python

8) تحليل المشاعر تجاه متحور Omicron باستخدام بايثون Omicron Sentiment Analysis using Python

قبل أيام قليلة، صنفت منظمة الصحة العالمية نوعًا جديداً من الفيروس التاجي (coronavirus)، B.1.1.529، كمتحور مثير للقلق أطلق عليه اسم Omicron. بعد ذلك مباشرة، رأينا انتشار التغريدات حول متحور Omicron على Twitter. لذا، إذاكنت تريد معرفة كيف يمكننا تحليل مشاعر التغريدات حول متحور Omicron، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل مشاعر Omicron باستخدام بايثون.

تحليل المشاعر لمتحور Omicron باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل مشاعر Omicron من Kaggle، والتي تم جمعهافي البداية من Twitter عندما كان الأشخاص يشاركون آرائهم حول متحور Omicron. فلنبدأ مهمة تحليل المشاعر Omicron عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data = pd.read_csv("omicron.csv")
print(data.head())
```

```
                    id            user_name  ... favorites is_retweet
0  1465693385088323591               Abaris  ...         0      False
1  1465693062999412746               GFTs ▯  ...         0      False
2  1465690116442279942  Herbie Finkle (Cozy)  ...         1      False
3  1465689607165591552    Electrical Review  ...         0      False
4  1465688203709464578      BingX Academy 🔑  ...         2      False

[5 rows x 16 columns]
```

مجموعة البيانات هذه كبيرة جداً، دعنا نلقي نظرة على ما إذاكانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

```
id                     0
user_name              0
user_location       4438
user_description    1278
user_created           0
user_followers         0
user_friends           0
user_favourites        0
user_verified          0
date                   0
text                   0
hashtags            4374
source                 0
retweets               0
favorites              0
is_retweet             0
dtype: int64
```

تحتوي مجموعة البيانات على قيم خاليةفي ثلاثة أعمدة تحتوي على بيانات نصية، وسأقوم بإزالة جميع الصفوف التي تحتوي على القيم الخالية :

```
data = data.dropna()
```

تحليل المشاعر لمتحور Omicron

يحتوي عمود النص (text column)في مجموعة البيانات على التغريدات التي قام بها الأشخاص لمشاركة آرائهم حول متحور Omicron. للمضي قدمًا، نحتاج إلى تنظيف هذا العمود وإعداده لمهمة تحليل المشاعر. إليك كيف يمكننا القيام بذلك:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
```

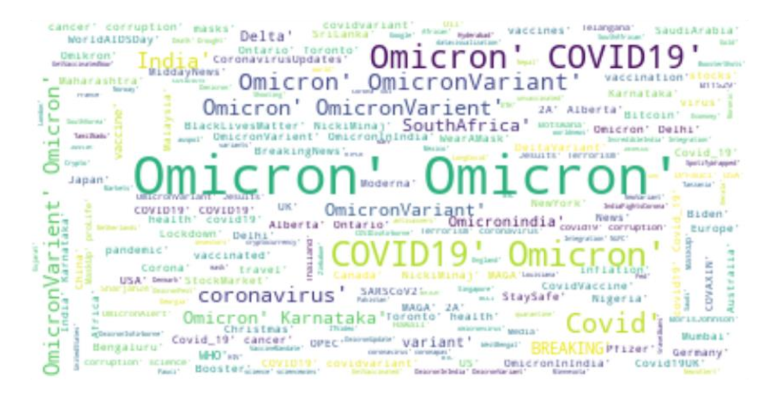
```
data["text"] = data["text"].apply(clean)
```

نظرًا لأننا قمنا بتنظيف عمود النص، فلنلقِ الآن نظرة على سحابة الكلمات (word cloud) في عمود النص لنلقي نظرة على أكبر عدد من الكلمات التي يستخدمها الأشخاص في تغريداتهم:

```
text = " ".join(i for i in data.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

دعنا الآن نلقي نظرة على سحابة الكلمات في عمود الوسوم (hashtags column) لإلقاء نظرة على أكبر عدد من علامات التصنيف التي يستخدمها الأشخاص في تغريداتهم:

```
text = " ".join(i for i in data.hashtags)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



الآن سأحسب درجات المشاعر للتغريدات حول متحور Omicron سأضيف هنا ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) في عمود النص:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["text"]]
data = data[["text", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                              text  Positive  Negative  Neutral
0  skynew told id back omicron "odium medicum ins...      0.16     0.000    0.840
1                          someon told octob omicron      0.00     0.000    1.000
3  autom system becom increas complex effort test...      0.00     0.000    1.000
5  digitaldisrupt emerg technolog stay privat inv...      0.00     0.000    1.000
7  fatigu head bodi ach occasion sore throat coug...      0.00     0.172    0.828
```

الآن دعونا نرى كيف كان رد فعل معظم الناس حول متحور Omicron :

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
  if (a>b) and (a>c):
    print("Positive 😊 ")
  elif (b>a) and (b>c):
    print("Negative 😠 ")
  else:
    print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك كانت معظم الآراء محايدة، مما يعني أن الأشخاص كانوا يشاركون معلومات حول متغير Omicron بدلاً من مشاركة أي آراء إيجابية أو سلبية.

الملخص

إذن هذه هي الطريقة التي يمكنك بها تحليل مشاعر متحور Omicron لفيروس كورونا. إنه نوع جديد من الفيروسات التاجية التي تم تصنيفها على أنها البديل المثير للقلق من قبل منظمة الصحة العالمية. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعر Omicron باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/12/06/omicron-sentiment-analysis-using-python/

9) تحليل المشاعر على Twitter باستخدام بايثون Twitter Sentiment Analysis using Python

Twitter هو أحد منصات التواصل الاجتماعي حيث يتمتع الأشخاص بحرية مشاركة آرائهم حول أي موضوع. نرى أحيانًا مناقشة قوية على Twitter حول رأي شخص ما تؤدي أحيانًا إلى مجموعة من التغريدات السلبية. مع وضع ذلك في الاعتبار، إذا كنت تريد معرفة كيفية إجراء تحليل المشاعر (sentiment analysis) على Twitter، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعر على Twitter باستخدام بايثون.

تحليل المشاعر على Twitter

تحليل المشاعر مهمة معالجة اللغة الطبيعية (natural language processing). يجب على جميع منصات وسائل التواصل الاجتماعي مراقبة مشاعر المشاركين في المناقشة. نرى في الغالب آراء سلبية على تويتر عندما تكون المناقشة سياسية. لذلك، يجب أن تستمر كل منصة في تحليل المشاعر للعثور على نوع الأشخاص الذين ينشرون الكراهية والسلبية على نظامهم الأساسي.

بالنسبة لمهمة تحليل المشاعر على Twitter، قمت بجمع مجموعة بيانات من Kaggle تحتوي على تغريدات حول مناقشة طويلة داخل مجموعة من المستخدمين. مهمتنا هنا هي تحديد عدد التغريدات السلبية والإيجابية حتى نتمكن من إعطاء نتيجة. لذلك،في القسم أدناه، سأقدم لك مهمة تحليل المشاعر على Twitter باستخدام بايثون.

تحليل المشاعر على Twitter باستخدام بايثون

لنبدأ مهمة تحليل المشاعر على Twitter من خلال استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import re
import nltk
import nltk

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/twitter.csv")
print(data.head())
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                               tweet
0  !!! RT @mayasolovely: As a woman you shouldn't t...
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
```

يحتوي عمود التغريدة (tweet column)في مجموعة البيانات أعلاه على التغريدات التي نحتاج إلى استخدامها لتحليل مشاعر المشاركين في المناقشة. ولكن للمضي قدمًا، يتعين علينا تنظيف الكثير من الأخطاء والرموز الخاصة الأخرى لأن هذه التغريدات تحتوي على الكثير من الأخطاء اللغوية. إذن إليك كيف يمكننا تنظيف عمود التغريدة:

```
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["tweet"] = data["tweet"].apply(clean)
```

الآن، الخطوة التالية هي حساب درجات المشاعر (sentiment scores) لهذه التغريدات وتعيين تسمية للتغريدات على أنها إيجابية (Posetive) أو سلبية (Negative) أو محايدة (Neutral). إليك كيفية حساب درجات المشاعرفي التغريدات:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["tweet"]]
```

```
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["tweet"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["tweet"]]
```

الآن سأختار فقط الأعمدة من هذه البيانات التي نحتاجها لبقية مهمة تحليل المشاعر على Twitter:

```
data = data[["tweet", "Positive",
             "Negative", "Neutral"]]
print(data.head())
```

```
                                             tweet  Positive  Negative  \
0  rt mayasolov woman shouldnt complain clean ho...     0.147     0.157
1  rt  boy dat coldtyga dwn bad cuffin dat hoe  ...     0.000     0.280
2  rt urkindofbrand dawg rt  ever fuck bitch sta...     0.000     0.577
3            rt cganderson vivabas look like tranni     0.333     0.000
4  rt shenikarobert shit hear might true might f...     0.154     0.407

   Neutral
0    0.696
1    0.720
2    0.423
3    0.667
4    0.440
```

دعنا الآن نلقي نظرة على التصنيف الأكثر شيوعًا المخصص للتغريدات وفقًا لدرجات المشاعر:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذا فإن معظم التغريدات محايدة (neutral)، ما يعني أنها ليست إيجابية ولا سلبية. الآن دعنا نلقي نظرة على إجمالي درجات المشاعر:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  2880.086000000009
Negative:  7201.020999999922
Neutral:  14696.887999999733
```

مجموع التغريدات المحايدة أعلى بكثير من السلبية والإيجابية، لكن من بين جميع التغريدات السلبية أكبر من التغريدات الإيجابية، لذلك يمكننا القول إن معظم الآراء سلبية.

الملخص

هذه هي الطريقة التي يمكنك بها أداء مهمة تحليل المشاعر على Twitter باستخدام لغة برمجة بايثون. تحليل المشاعر مهمة معالجة اللغة الطبيعية. تحتاج جميع منصات وسائل التواصل الاجتماعي إلى التحقق من مشاعر الأشخاص المشاركين في المناقشة. آمل أن تكون قد أحببت هذا المقال على تحليل المشاعر على Twitter باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/09/13/twitter-sentiment-/analysis-using-python

10) تحليل مشاعر لعبة الحبار باستخدام بايثون Squid Game Sentiment Analysis using Python

تعد لعبة الحبار (squid game) حاليًا واحدة من أكثر العروض شيوعًا على Netflix. من الشائع جدًا أن الأشخاص الذين لم يشاهدوا أي سلسلة ويب من قبل يشاهدونها أيضًا. أحد أسباب ذلك هو آراء وآراء المشاهدين على وسائل التواصل الاجتماعي. لذلك إذا كنت تريد معرفة كيفية تحليل مشاعر الناس حول لعبة الحبار، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تحليل مشاعر لعبة الحبار باستخدام بايثون.

تحليل المشاعر لعبة الحبار باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل مشاعر لعبة الحبار من Kaggle، والتي تم جمعهافي البداية من Twitter بينما كان الأشخاص يشاركون بنشاط آرائهم حول لعبة الحبار. لنبدأ مهمة تحليل مشاعر لعبة الحبار عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data = pd.read_csv("squid_game.csv")
print(data.head())
```

```
                     user_name  user_location  ...               source is_retweet
0   the _ûndër-ratèd niggáh👊🏾            NaN  ...  Twitter for Android      False
1  Best uncle on planet earth            NaN  ...  Twitter for Android      False
2                       marcie            NaN  ...      Twitter Web App      False
3                     YoMo.Mdp  Any pronouns   ...      Twitter Web App      False
4               Laura Reactions         France  ...      Twitter Web App      False

[5 rows x 12 columns]
```

في الانطباعات الأولى لمجموعة البيانات هذه، لاحظت قيمًا خاليةفي عمود (user_location) يبدو أنها لا تؤثر على مهمة تحليل المشاعر. لذلك سوف احذف هذا العمود:

```
data = data.drop(columns="user_location", axis=1)
```

دعنا الآن نلقي نظرة على ما إذا كانت الأعمدة الأخرى تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

```
user_name             4
user_description   5211
user_created          0
user_followers        0
user_friends          0
user_favourites       0
user_verified         0
date                  0
text                  0
source                0
is_retweet            0
dtype: int64
```

يحتوي عمود (user_description) أيضًا على قيم خالية، والتي لن تؤثر أيضًا على مهمة تحليل المشاعر. لذلك سأحذف هذا العمود أيضًا:

```
data = data.drop(columns="user_description", axis=1)
data = data.dropna)
```

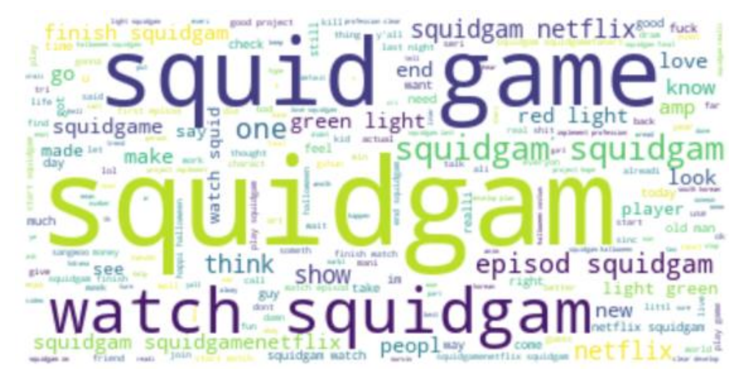
يحتوي عمود النص (text)في مجموعة البيانات على آراء مستخدمي تويتر حول لعبة الحبار، فهذه آراء لوسائل التواصل الاجتماعي، لذا يجب تحضير هذا العمود قبل أي تحليل. لذلك دعونا نجهز هذا العمود لمهمة تحليل المشاعر:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["text"] = data["text"].apply(clean)
```

دعنا الآن نلقي نظرة على الكلمات الأكثر استخدامًافي آراء لعبة الحبار باستخدام سحابة الكلمات (word cloud). سحابة الكلمات هي أداة تصوير البيانات تعرض الكلمات الأكثر استخدامًا بحجم أكبر. إليك كيفية تصوير سحابة الكلماتفي عمود النص:

```
text = " ".join(i for i in data.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



الآن دعنا ننتقل إلى مهمة تحليل المشاعرفي لعبة الحبار. سأضيف هنا ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores)في عمود النص:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["text"]]
data = data[["text", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                            text  Positive  Negative  Neutral
0  life hit time poverti strike yougong yoo  let ...     0.173     0.108    0.719
1                   marbl episod squidgam  ruin 😭😭😭     0.000     0.487    0.513
2                                   squidgam time     0.000     0.000    1.000
3  blood  slideim join squidgam thing im alreadi ...     0.142     0.277    0.581
4  two first game player kill mask guy  bloodi ni...     0.000     0.461    0.539
```

دعنا الآن نحسب كيف يفكر معظم الناس في لعبة الحبار:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
```

```
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذا فإن معظم آراء المستخدمين حيادية، فلنلقِ الآن نظرة على إجمالي كل درجة المشاعر قبل التوصل إلى أي استنتاج:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  10604.55899999976
Negative:  5171.334000000031
Neutral:  64233.11800000302
```

إجمالي السلبيات أقل بكثير من الإيجابية، لذلك يمكننا القول إن معظم الآراء حول لعبة الحبار إيجابية.

الملخص

تعد لعبة الحبار حاليًا واحدة من أكثر العروض شيوعًا على Netflix. أحد أسباب ذلك هو آراء وآراء المشاهدين على وسائل التواصل الاجتماعي. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعرفي لعبة الحبار باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/11/03/squid-game-sentiment-/analysis-using-python

11) تحليل المشاعر في دردشة WhatsApp باستخدام بايثون

WhatsApp Chat Sentiment Analysis using Python

يعد WhatsApp مصدرًا رائعًا للبيانات لتحليل العديد من الأنماط والعلاقات بين شخصين أو أكثر يتحدثون شخصيًا أو حتى في مجموعات. إذا كنت تريد أن تعرف كيف يمكننا تحليل مشاعر دردشة WhatsApp، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعرفي دردشة WhatsApp باستخدام بايثون.

تحليل المشاعر دردشة WhatsApp

لتحليل مشاعر دردشة WhatsApp، نحتاج إلى جمع البيانات من WhatsApp. يجب أن يستخدم معظمكم تطبيق المراسلة هذا، لذلك لجمع البيانات حول الدردشة، ما عليك سوى اتباع الخطوات المذكورة أدناه:

1. **للآيفون:**
    1. افتح الدردشة الخاصة بك مع شخص أو مجموعة.
    2. فقط اضغط على ملف تعريف الشخص أو المجموعة.
    3. سترى خيارًا لتصدير الدردشة بالأسفل.
4. **لأجهزة الأندرويد:**
    1. افتح الدردشة الخاصة بك مع شخص أو مجموعة.
    2. انقر فوق النقاط الثلاث أعلاه.
    3. انقر فوق المزيد.
    4. انقر فوق دردشة التصدير.

سترى خيارًا لإرفاق الوسائط أثناء تصدير الدردشة. من أجل البساطة، من الأفضل عدم إرفاق الوسائط. أخيرًا، أدخل بريدك الإلكتروني وستجد دردشة WhatsApp في صندوق الوارد الخاص بك.

تحليل مشاعر الدردشة عبر WhatsApp باستخدام بايثون

لنبدأ الآن بمهمة تحليل مشاعر الدردشة عبر WhatsApp باستخدام بايثون. سأبدأ هذه المهمة بتحديد بعض الدوال المساعدة لأن البيانات التي نحصل عليها من WhatsApp ليست

مجموعة بيانات جاهزة للاستخدام في أي نوع من مهام علم البيانات. لذلك، لإعداد بياناتك لمهمة تحليل المشاعر، ما عليك سوى تحديد جميع الدوال على النحو المحدد أدناه:

```
import re
import pandas as pd
import numpy as np
import emoji
from collections import Counter
import matplotlib.pyplot as plt
from PIL import Image
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

# Extract Time
def date_time(s):
    pattern = '^([0-9]+)(\/)([0-9]+)(\/)([0-9]+), ([0-
9]+):([0-9]+)[ ]?(AM|PM|am|pm)? -'
    result = re.match(pattern, s)
    if result:
        return True
    return False

# Find Authors or Contacts
def find_author(s):
    s = s.split(":")
    if len(s)==2:
        return True
    else:
        return False

# Finding Messages
def getDatapoint(line):
    splitline = line.split(' - ')
    dateTime = splitline[0]
    date, time = dateTime.split(", ")
    message = " ".join(splitline[1:])
    if find_author(message):
        splitmessage = message.split(": ")
        author = splitmessage[0]
        message = " ".join(splitmessage[1:])
    else:
        author= None
    return date, time, author, message
```

لا يهم إذا كنت تستخدم مجموعة بيانات دردشة جماعية أو محادثتك مع شخص واحد. ستعمل جميع الدوال المحددة أعلاه على إعداد بياناتك لمهمة تحليل المشاعر وأي مهمة تتعلق بعلم البيانات. الآن إليك كيف يمكننا إعداد البيانات التي جمعناها من WhatsApp باستخدام الدوال المذكورة أعلاه:

```
data = []
conversation = 'WhatsApp Chat with Sapna.txt'
with open(conversation, encoding="utf-8") as fp:
    fp.readline()
```

```
    messageBuffer = []
    date, time, author = None, None, None
    while True:
        line = fp.readline()
        if not line:
            break
        line = line.strip()
        if date_time(line):
            if len(messageBuffer) > 0:
                data.append([date, time, author, '
'.join(messageBuffer)])
            messageBuffer.clear()
            date, time, author, message = getDatapoint(line)
            messageBuffer.append(message)
        else:
            messageBuffer.append(line)
```

الآن إليك كيف يمكننا تحليل مشاعر دردشة WhatsApp باستخدام بايثون:

```
df = pd.DataFrame(data, columns=["Date", 'Time', 'Author',
'Message'])
df['Date'] = pd.to_datetime(df['Date'])

data = df.dropna()
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Message"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Message"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Message"]]
print(data.head())
```

```
         Date       Time         Author  ... Positive  Negative  Neutral
0  2020-04-06   12:30 pm          Sapna  ...      0.0     0.000    1.000
1  2020-04-06   12:30 pm          Sapna  ...      0.0     0.000    1.000
2  2020-04-06   12:54 pm   Aman Kharwal  ...      0.0     0.000    1.000
3  2020-04-06   12:55 pm          Sapna  ...      0.0     0.383    0.617
4  2020-04-06    1:00 pm   Aman Kharwal  ...      0.0     0.000    1.000
```

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Output:
Neutral 🙂
```

لذا، فإن البيانات التي استخدمتها تشير إلى أن معظم الرسائل بيني وبين الشخص الآخر محايدة (neutral). مما يعني أنه ليس إيجابيًا (positive) أو سلبيًا (negative).

الملخص

إذن هذه هي الطريقة التي يمكننا بها أداء مهمة تحليل المشاعرفي دردشة WhatsApp. يعد WhatsApp مصدرًا رائعًا للبيانات لمهمة تحليل المشاعر وكل مهمةفي علم البيانات تعتمد على معالجة اللغة الطبيعية. آمل أن تكون قد أحببت هذه المقالة حول مهمة تحليل المشاعرفي دردشة WhatsApp باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/06/06/whatsapp-chat-/sentiment-analysis-using-python

12) تحليل المشاعر في تقييمات منتجات Amazon باستخدام بايثون

Amazon Product Reviews Sentiment Analysis with Python

Amazon هي شركة أمريكية متعددة الجنسيات تركز على التجارة الإلكترونية والحوسبة السحابية والبث الرقمي ومنتجات الذكاء الاصطناعي. لكنها معروفة بشكل أساسي بمنصة التجارة الإلكترونية الخاصة بها والتي تعد واحدة من أكبر منصات التسوق عبر الإنترنت اليوم. هناك الكثير من العملاء الذين يشترون المنتجات من Amazon حتى أن Amazon تكسب اليوم في المتوسط 638.1 مليون دولار في اليوم. لذا، فإن وجود مثل هذه القاعدة الكبيرة من العملاء، سيصبح مشروعًا رائعًا لعلم البيانات إذا تمكنا من تحليل مشاعر مراجعات منتجات Amazon. لذلك، في هذه المقالة، سوف أطلعك على مهمة تحليل آراء مراجعات منتجات Amazon باستخدام بايثون.

تحليل المشاعر في تقييمات منتجات Amazon باستخدام بايثون

تم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل آراء مراجعات منتجات Amazon من Kaggle. تحتوي مجموعة البيانات هذه على مراجعات المنتجات لأكثر من 568000 عميل اشتروا منتجات من Amazon. فلنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()

data = pd.read_csv("Reviews.csv")
print(data.head())
```

```
   Id   ProductId          UserId                      ProfileName  \
0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"
3   4  B000UA0QIQ  A395BORC6FGVXV                             Karl
4   5  B006K2ZZ7K  A1UQRSCLF8GW1T    Michael D. Bigham "M. Wassir"

   HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                     1                       1      5  1303862400
1                     0                       0      1  1346976000
2                     1                       1      4  1219017600
3                     3                       3      2  1307923200
4                     0                       0      5  1350777600

                Summary                                               Text
0 Good Quality Dog Food  I have bought several of the Vitality canned d...
1     Not as Advertised  Product arrived labeled as Jumbo Salted Peanut...
2 "Delight" says it all  This is a confection that has been around a fe...
3        Cough Medicine  If you are looking for the secret ingredient i...
4           Great taffy  Great taffy at a great price.  There was a wid...
```

قبل المضي قدمًا، دعنا نلقي نظرة على بعض المعلومات المطلوبة من مجموعة البيانات هذه:

```
print(data.describe())
```

```
                 Id  HelpfulnessNumerator  HelpfulnessDenominator  \
count  568454.000000         568454.000000            568454.00000
mean   284227.500000              1.743817                 2.22881
std    164098.679298              7.636513                 8.28974
min         1.000000              0.000000                 0.00000
25%    142114.250000              0.000000                 0.00000
50%    284227.500000              0.000000                 1.00000
75%    426340.750000              2.000000                 2.00000
max    568454.000000            866.000000               923.00000

               Score          Time
count  568454.000000  5.684540e+05
mean        4.183199  1.296257e+09
std         1.310436  4.804331e+07
min         1.000000  9.393408e+08
25%         4.000000  1.271290e+09
50%         5.000000  1.311120e+09
75%         5.000000  1.332720e+09
max         5.000000  1.351210e+09
```

نظرًا لأن مجموعة البيانات هذه كبيرة جداً، فهي تحتوي على بعض القيم المفقودة (missing values)، لذا دعنا نزيل جميع الصفوف التي تحتوي على القيم المفقودة:

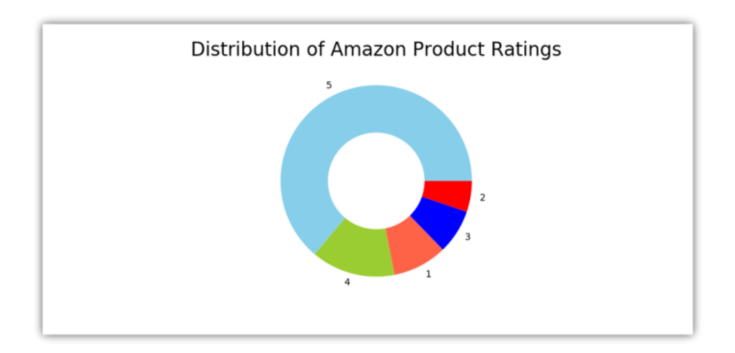
```
data = data.dropna()
```

تحليل المشاعر لمراجعات منتجات Amazon

يحتوي عمود النقاط (Score column) في مجموعة البيانات هذه على التصنيفات التي منحها العملاء للمنتج بناءً على تجربتهم مع المنتج. لذلك دعونا نلقي نظرة على تفاصيل التصنيف لمعرفة كيف يقوم معظم العملاء بتقييم المنتجات التي يشترونها من Amazon:

```
ratings = data["Score"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(10, 8))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Distribution of Amazon Product Ratings",
fontsize=20)
plt.show()
```



وفقًا للشكل أعلاه، صنف أكثر من نصف الأشخاص المنتجات التي اشتروها من Amazon بـ 5 نجوم، وهو أمر جيد. الآن، سأضيف ثلاثة أعمدة أخرى إلى مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) لمراجعات العملاء المذكورة في عمود النص (Text column) لمجموعة البيانات:

```
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Text"]]
print(data.head())
```

```
   Id   ProductId          UserId  ... Positive  Negative  Neutral
0   1  B001E4KFG0  A3SGXH7AUHU8GW  ...    0.305     0.000    0.695
1   2  B00813GRG4  A1D87F6ZCVE5NK  ...    0.000     0.138    0.862
2   3  B000LQOCH0   ABXLMWJIXXAIN  ...    0.155     0.091    0.754
3   4  B000UA0QIQ  A395BORC6FGVXV  ...    0.000     0.000    1.000
4   5  B006K2ZZ7K  A1UQRSCLF8GW1T  ...    0.448     0.000    0.552

[5 rows x 13 columns]
```

دعنا الآن نرى كيف قام معظم الناس بتقييم المنتجات التي اشتروها من أمازون:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
```

```
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك، يكون معظم الأشخاص محايدين عند إرسال تجاربهم مع المنتجات التي اشتروها من Amazon. الآن دعنا نرى إجمالي جميع درجات المشاعر :

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  109328.1269999992
Negative:  24033.022999999564
Neutral:   435043.95799998916
```

لذلك يمكننا القول إن معظم المراجعات للمنتجات المتاحة على Amazon إيجابية، حيث أن إجمالي درجات المشاعر الإيجابية والمحايدة أعلى بكثير من النتائج السلبية.

الملخص

إذن هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات المنتج في Amazon. هناك الكثير من العملاء الذين يشترون المنتجات من Amazon حتى أن Amazon تكسب اليوم في المتوسط 638.1 مليون دولار في اليوم. لذا، فإن وجود مثل هذه القاعدة الكبيرة من العملاء، سيصبح مشروعًا رائعًا لعلوم البيانات إذا تمكنا من تحليل مشاعر مراجعات منتجات Amazon. آمل أن تكون قد أحببت هذه المقالة حول تحليل المشاعر لمراجعات Amazon باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/07/20/amazon-product-/reviews-sentiment-analysis-with-python

13) تحليل مشاعر تقييمات الفندق مع بايثون Hotel Reviews Sentiment Analysis with Python

عندما نبحث عن فنادق لقضاء الإجازة أو السفر، نفضل دائمًا فندقًا معروفًا بخدماته. أفضل طريقة لمعرفة ما إذا كان الفندق مناسبًا لك أم لا هي معرفة ما يقوله الناس عن الفندق الذي أقام هناك من قبل. من الصعب للغاية الآن قراءة تجربة كل شخص أبدى رأيهفي خدمات الفندق. هذا هو المكان الذي تأتي فيه مهمة تحليل المشاعر (sentiment analysis).في هذه المقالة، سوف أطلعك على مهمة تحليل مشاعر تقييمات الفندق (Hotel Reviews Sentiment Analysis) باستخدام بايثون.

تحليل مشاعر تقييمات الفندق مع بايثون

يتم جمع مجموعة البيانات التي أستخدمها لمهمة تحليل آراء الفنادق من Kaggle. يحتوي على بيانات حول 20000 تقييم للأشخاص حول خدمات الفنادق التي أقاموا فيها لقضاء عطلة أو رحلة عمل أو أي نوع من الرحلات. تحتوي مجموعة البيانات هذه على عمودين فقط كمراجعات (Reviews) وتقييمات (Ratings) للعملاء. لذا فلنبدأ بمهمة تحليل آراء الفنادق باستخدام بايثون من خلال استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()

data = pd.read_csv("hotel_reviews.csv")
print(data.head())
```

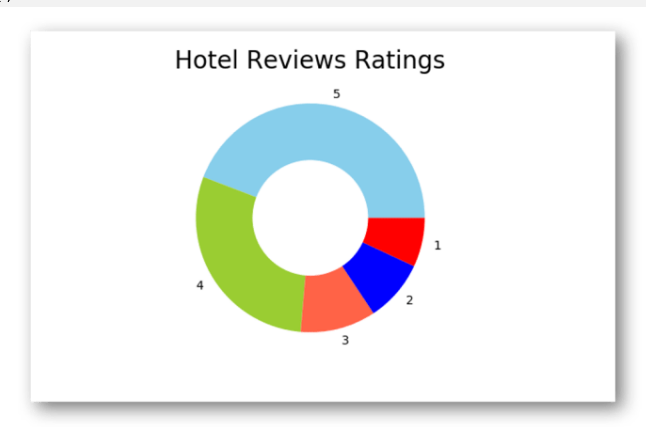
```
                                              Review  Rating
0  nice hotel expensive parking got good deal sta...       4
1  ok nothing special charge diamond member hilto...       2
2  nice rooms not 4* experience hotel monaco seat...       3
3  unique, great stay, wonderful time hotel monac...       5
4  great stay great stay, went seahawk game aweso...       5
```

مجموعة البيانات هذه كبيرة جدًا ولحسن الحظ لا توجد قيم مفقودة، لذا دون إضاعة أي وقت، دعنا نلقي نظرة سريعة على توزيع تقييمات العملاء:

```
ratings = data["Rating"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(5, 5))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
```

```
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Hotel Reviews Ratings", fontsize=20)
plt.show()
```



يمكن ملاحظة أن معظم النزلاء قيموا الخدمات الفندقية بـ 5 نجوم و4 نجوم. لذلك وفقًا للتصنيفات المذكورة أعلاه، يمكننا القول إن معظم الضيوف راضون عن خدمات الفندق الذي أقاموا فيه. دعنا الآن نمضي قدمًا من خلال تحليل مشاعر تقييمات الفنادق. لتحليل وجهة نظر تقييمات الفندق، سأضيف ثلاثة أعمدة إضافية إلى مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) للتقييمات:

```
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Review"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Review"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Review"]]
print(data.head())
```

Review	Rating	Positive	Positive	Negative	Neutral
nice hotel expensive parking got good deal sta...	4	0.285	0.285	0.072	0.643
ok nothing special charge diamond member hilto...	2	0.189	0.189	0.110	0.701
nice rooms not 4* experience hotel monaco seat...	3	0.219	0.219	0.081	0.700
unique, great stay, wonderful time hotel monac...	5	0.385	0.385	0.060	0.555
great stay great stay, went seahawk game aweso...	5	0.221	0.221	0.135	0.643

وفقًا للتقييمات (reviews)، يبدو أن ضيوف الفندق راضون عن الخدمات، فلنلقِ الآن نظرة على رأي معظم الناس في خدمات الفنادق بناءً على مشاعر تقييمهم:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])
```

```
def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

وبالتالي، يشعر معظم الناس بالحياد (neutral) تجاه خدمات الفندق. الآن دعنا نلقي نظرة فاحصة على نتائج المشاعر:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  6359.91000000002
Negative:  1473.4750000000038
Neutral:  12657.627999999937
```

وبالتالي، وفقًا للنتائج المذكورة أعلاه، تم تصنيف أكثر من 12000 مراجعة على أنها محايدة، وتم تصنيف أكثر من 6000 مراجعة على أنها إيجابية. لذلك يمكن القول إن الناس سعداء حقًا بخدمات الفنادق التي أقاموا فيها حيث أن التقييمات السلبية أقل من 1500.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل مشاعر تقييمات الفنادق. أفضل طريقة لمعرفة ما إذا كان الفندق مناسبًا لك أم لا هي معرفة ما يقوله الناس عن الفندق الذي أقام هناك من قبل. هذا هو المكان الذي يمكن أن تساعدك فيه مهمة تحليل مشاعر تقييمات الفندق على تحديد ما إذا كان الفندق مناسبًا لرحلتك أم لا. آمل أن تكون قد أحببت هذه المقالة حول تحليل المشاعر لتقييمات الفنادق باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/07/03/hotel-reviews-/sentiment-analysis-with-python

14) تحليل المشاعر في متجر Google Play باستخدام بايثون Google Play Store Sentiment Analysis using Python

تحليل المشاعر (Sentiment analysis) هو تصنيف لمراجعات العميل أو تعليقاته على أنها إيجابية (positive) وسلبية (negative) وأحيانًا محايدة (neutral) أيضًا. تحلل معظم الأنشطة التجارية مشاعر عملائها حول منتجاتهم أو خدماتهم لمعرفة ما يريده عملاؤهم منهم. يحتوي متجر Google play على ملايين التطبيقات مع مراجعاتها، لذا ستكون حالة استخدام جيدة لتحليل المشاعر لتحليل مشاعر التطبيقات المتاحة على متجر Google play. لذلك،في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعرفي متجر Google Play باستخدام بايثون.

تحليل المشاعر في متجر Google Play

يمكن العثور على متجر Google Play على جميع الهواتف الذكية والأجهزة اللوحية التي تعمل بنظام Android. هذا هو متجر تطبيقات Google الرسمي لنظام التشغيل Android. يحتوي على ملايين التطبيقات مع مراجعاتها حتى نتمكن من استخدام مثل هذا القدر من البيانات لأي مهمة تتعلق بعلم البيانات. تحليل آراء العملاء وتعليقاتهم هو ما نقوم بهفي مهمة تحليل المشاعر. بعد قولي هذا،في القسم أدناه، سوف أطلعك على مهمة تحليل المشاعرفي متجر Google Play باستخدام بايثون. يمكن تنزيل مجموعة البيانات التي أستخدمهافي هذه المهمة من هنا.

تحليل المشاعر في متجر Google Play باستخدام بايثون

سأبدأ هذه المهمة بقراءة مجموعة البيانات. يتم تنزيل مجموعة البيانات التي أستخدمها هنا من Kaggle والتي تم جمعها من متجر Google Play. فلنبدأ هذه المهمة بقراءة مجموعة البيانات:

```
from itertools import count
from nltk.util import pr
import pandas as pd
data = pd.read_csv("user_reviews.csv")
print(data.head())
```

```
                   App  ... Sentiment_Subjectivity
0  10 Best Foods for You  ...               0.533333
1  10 Best Foods for You  ...               0.288462
2  10 Best Foods for You  ...                    NaN
3  10 Best Foods for You  ...               0.875000
4  10 Best Foods for You  ...               0.300000
```

قبل المضي قدمًا، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم مفقودة أم لا:

```
print(data.isnull().sum())
```

```
App                          0
Translated_Review        26868
Sentiment                26863
Sentiment_Polarity       26863
Sentiment_Subjectivity   26863
dtype: int64
```

إذاً تحتوي على بعض القيم الفارغة، سأقوم بإنشاء مجموعة بيانات جديدة بإسقاط القيم الخالية:

```
data = data.dropna()
print(data.isnull().sum())
```

```
App                       0
Translated_Review         0
Sentiment                 0
Sentiment_Polarity        0
Sentiment_Subjectivity    0
dtype: int64
```

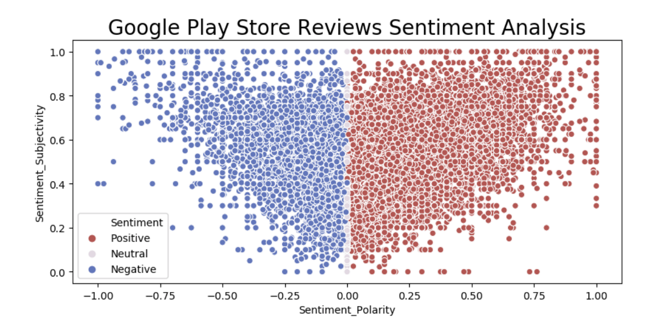
الآن لتحليل المشاعر الخاصة بمراجعات متجر google play ، سأضيف ثلاثة أعمدة جديدةفي مجموعة البيانات من خلال فهم مشاعر كل مراجعة للعميل على أنها إيجابية وسلبية ومحايدة:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Translated_Review"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Translated_Review"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Translated_Review"]]
print(data.head())
```

```
                   App                         Translated_Review  ... Negative  Neutral
0  10 Best Foods for You  I like eat delicious food. That's I'm cooking ...  ...      0.0    0.466
3  10 Best Foods for You       Works great especially going grocery store  ...      0.0    0.549
4  10 Best Foods for You                                  Best idea us  ...      0.0    0.323
5  10 Best Foods for You                                      Best way  ...      0.0    0.192
```

والآن كخطوة أخيرة، دعنا نلقي نظرة على مشاعر العملاء حول التطبيقات المتاحةفي متجر Google play باستخدام مخطط مبعثر (scatter plot):

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(15, 10))
sns.scatterplot(data['Sentiment_Polarity'],
data['Sentiment_Subjectivity'],
                hue = data['Sentiment'], edgecolor='white',
palette="twilight_shifted_r")
plt.title("Google Play Store Reviews Sentiment Analysis",
fontsize=20)
plt.show()
```



الملخص

إذن هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات متجر google play تحليل المشاعر هو تصنيف لمراجعات العميل أو تعليقاته على أنها إيجابية وسلبية ومحايدة. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة تحليل مشاعر متجر Google play باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/05/31/google-play-store-/sentiment-analysis-using-python

15) تحليل مشاعر مراجعات Amazon Alexa باستخدام بايثون Amazon Alexa Reviews Sentiment Analysis using Python

Amazon Alexa هي خدمة صوتية مستندة إلى السحابة تم تطويرها بواسطة Amazon تتيح للعملاء التفاعل مع التكنولوجيا. يوجد حاليًا أكثر من 40 مليون مستخدم لـ Alexa حول العالم، لذا فإن تحليل مشاعر المستخدمين حول Alexa سيكون مشروعًا جيداً لعلم البيانات. لذا، إذا كنت تريد معرفة كيفية تحليل مشاعر المستخدمين باستخدام Amazon Alexa، فهذه المقالة مناسبة لك.في هذه المقالة، سأوجهك خلال مهمة تحليل آراء Amazon Alexa باستخدام بايثون.

تحليل مشاعر مراجعات Amazon Alexa باستخدام بايثون

تم جمع مجموعة البيانات التي أستخدمها لمهمة تحليل المشاعر لمراجعات Amazon Alexa من Kaggle. يحتوي على بيانات حول التصنيفات بين 1 و 5، وتاريخ المراجعات ، وتعليقات العملاء حول تجربتهم مع Alexa. لذلك دعونا نستورد مجموعة بيانات ومكتبات بايثون الضرورية التي نحتاجها لهذه المهمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/amazon_alexa.tsv", delimiter='\t')
print(data.head())
```

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1

لنبدأ بإلقاء نظرة على بعض المعلومات الموجودةفي تلك البيانات لمعرفة ما إذا كنا بحاجة إلى تغييرها أم لا:

```
print(data.describe())
print(data.isnull().sum())
print(data.columns)
```

يحتوي عمود تصنيف مجموعة البيانات (dataset's rating column) على التقييمات التي قدمها مستخدمو Amazon Alexa بمقياس من 1 إلى 5، حيث يمثل الرقم 5 أفضل تقييم.

```
               rating      feedback
count     3150.000000   3150.000000
mean         4.463175      0.918413
std          1.068506      0.273778
min          1.000000      0.000000
25%          4.000000      1.000000
50%          5.000000      1.000000
75%          5.000000      1.000000
max          5.000000      1.000000
rating                0
date                  0
variation             0
verified_reviews      0
feedback              0
dtype: int64
Index(['rating', 'date', 'variation', 'verified_reviews', 'feedback'], dtype='object')
```

يمكن للمستخدم تقديمه. لذلك دعونا نلقي نظرة على توزيع التقييمات التي منحها مستخدمو Amazon Alexa:

```
ratings = data["rating"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(5, 5))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Amazon Alexa Reviews", fontsize=20)
plt.show()
```



من الشكل أعلاه، يمكننا أن نرى أن معظم العملاء قد صنفوا Amazon Alexa بما في ذلك جميع متغيراتها على أنها 5. لذا فهذا يعني أن معظم العملاء سعداء بخدمة Amazon Alexa.

تحليل مشاعر مراجعات Amazon Alexa

الآن دعنا ننتقل إلى مهمة تحليل المشاعر لمراجعات Alexa. يحتوي عمود المراجعات التي تم التحقق منها (verified_reviews column)في مجموعة البيانات على جميع المراجعات التي قدمها عملاء Amazon Alexa. لذلك دعونا نضيف أعمدة جديدة إلى هذه البيانات على أنها أعمدة إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر للمراجعات (sentiment scores):

```
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["verified_reviews"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["verified_reviews"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["verified_reviews"]]
print(data.head())
```

```
   rating       date         variation  ... Positive  Negative  Neutral
0       5  31-Jul-18  Charcoal Fabric   ...    0.692     0.000    0.308
1       5  31-Jul-18  Charcoal Fabric   ...    0.807     0.000    0.193
2       4  31-Jul-18    Walnut Finish   ...    0.114     0.102    0.784
3       5  31-Jul-18  Charcoal Fabric   ...    0.383     0.000    0.617
4       5  31-Jul-18  Charcoal Fabric   ...    0.000     0.000    1.000
```

دعنا الآن نلخص درجات المشاعر لكل عمود لفهم ما يعتقده معظم عملاء Amazon Alexa بشأنه:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

وبالتالي، فإن الناتج النهائي الذي نحصل عليه يكون محايداً. هذا يعني أن معظم المستخدمين يشعرون بالحياد تجاه خدمات Amazon Alexa. دعنا الآن نرى مجموع درجات المشاعر لكل عمود:

```
print("Positive: ", x)
print("Negative: ", y)
```

```
print("Neutral: ", z)
```

```
Positive:   1035.4579999999983
Negative:   96.79999999999995
Neutral:   1936.740999999996
```

لذلك يمكننا أن نرى أن الايجابي والمحايد أعلى من 1000 حيث يكون السلبي أقل من 100. وهذا يعني أن معظم عملاء Amazon Alexa راضون عن خدماتها.

الملخص

هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات Amazon Alexa باستخدام لغة برمجة بايثون. يوجد حاليًا أكثر من 40 مليون مستخدم لـ Alexa حول العالم، لذا فإن تحليل مشاعر المستخدمين حول Alexa سيكون مشروعًا جيداً لعلوم البيانات. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة تحليل آراء Amazon Alexa باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/06/12/amazon-alexa-reviews-/sentiment-analysis-using-python

16) التنبؤ بالانتخابات الأمريكية باستخدام لغة بايثون Predict US Elections with Python

في هذه المقالة، سأطلعك على كيفية التنبؤ بالانتخابات الرئاسية الأمريكية باستخدام بايثون. هنا، لن أقوم بتدريب أي نموذج للتعلم الآلي. سوف أقوم بتحليل مشاعر الناس تجاه المرشحين وبعد ذلكفي النهاية، سأختتم بناءً على أكبر عدد من التغريدات الإيجابية والسلبية ضد المرشحين.

تم جمع مجموعات البيانات التي أستخدمهافي هذه المهمة للتنبؤ بالانتخابات الأمريكية من Twitter بواسطة تويتر هاندلز twitter handles الرسمية لدونالد ترامب وجو بايدن. يمكنك تنزيل مجموعات البيانات التي أستخدمها من هنا.

التنبؤ بالانتخابات الأمريكية باستخدام لغة بايثون

الآن دون إضاعة أي وقت، فلنبدأ بهذه المهمة للتنبؤ بالانتخابات الأمريكية باستخدام بايثون عن طريق استيراد المكتبات ومجموعات البيانات اللازمة:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from textblob import TextBlob
from wordcloud import WordCloud
import plotly.graph_objects as go
import plotly.express as px

trump_reviews = pd.read_csv("Trumpall2.csv")
biden_reviews = pd.read_csv("Bidenall2.csv")
```

دعنا الآن نلقي نظرة سريعة على الصفوف الخمسة الأولى من مجموعتي البيانات:

```
print(trump_reviews.head())
print(biden_reviews.head())
```

```
           user             text
0      manny_rosen    @sanofi please tell us how many shares the Cr...
1        osi_abdul    https://t.co/atM98CpqF7  Like, comment, RT #P...
2          Patsyrw    Your AG Barr is as useless & corrupt as y...
3  seyedebrahimi_m    Mr. Trump! Wake Up!  Most of the comments bel...
4     James09254677   After 4 years you think you would have figure...
```

```
        user              text
0   MarkHodder3    @JoeBiden And we’ll find out who won in 2026...
1    K87327961G   @JoeBiden Your Democratic Nazi Party cannot be...
2      OldlaceA                         @JoeBiden So did Lying Barr
3    penblogger   @JoeBiden It's clear you didnt compose this tw...
4  Aquarian0264          @JoeBiden I will vote in person thank you.
```

تحليل المشاعر

الآن، سأبدأ بتحليل المشاعر. سأفعل ذلك باستخدام حزمة Textblob في بايثون. سأستخدم هنا هذه الحزمة لأداء تصنيف بسيط للنص إما إيجابيًا أو سلبيًا على أساس تحليل المشاعر:

```
textblob1 = TextBlob(trump_reviews["text"][10])
print("Trump :",textblob1.sentiment)
textblob2 = TextBlob(biden_reviews["text"][500])
print("Biden :",textblob2.sentiment)
```

Trump : Sentiment(polarity=0.15, subjectivity=0.3125)

Biden : Sentiment(polarity=0.6, subjectivity=0.9)

```
def find_pol(review):
    return TextBlob(review).sentiment.polarity
trump_reviews["Sentiment Polarity"] =
trump_reviews["text"].apply(find_pol)
print(trump_reviews.tail())

biden_reviews["Sentiment Polarity"] =
biden_reviews["text"].apply(find_pol)
print(biden_reviews.tail())
```

```
                 user  ...      Sentiment Polarity
2783          4diva63  ...                   0.000
2784         hidge826  ...                   0.000
2785      SpencerRossy  ...                  0.225
2786  ScoobyMcpherson  ...                   0.000
2787          bjklinz  ...                  -0.500
```

```
             user  ...      Sentiment Polarity
2535    meryn1977  ...                  0.15
2536  BSNelson114  ...                  0.00
2537     KenCapel  ...                  0.00
2538   LeslyeHale  ...                  0.10
2539     rerickre  ...                  0.20
```

الآن قبل المضي قدمًا، دعونا نفهم ما هو القطبية Polarity. تتراوح القطبية من ـ1 إلى +1 (سلبي إلى إيجابي) وتخبر ما إذا كان النص يحتوي على مشاعر سلبية أو مشاعر إيجابية. تخبر القطبية عن المعلومات الواقعية.

استقطاب المشاعر على كلا المرشحين:

الآن سأضيف سمة جديدة في كلتا مجموعتي البيانات باسم "Expression Label":

```
trump_reviews["Expression Label"] =
np.where(trump_reviews["Sentiment Polarity"]>0, "positive",
"negative")
trump_reviews["Expression Label"][trump_reviews["Sentiment
Polarity"]==0]="Neutral"
```

```
print(trump_reviews.tail())

biden_reviews["Expression Label"] =
np.where(biden_reviews["Sentiment Polarity"]>0, "positive",
"negative")
biden_reviews["Expression Label"][trump_reviews["Sentiment
Polarity"]==0]="Neutral"
print(biden_reviews.tail())
```

الآن سأقوم بإسقاط جميع التغريدات ذات القطبية المحايدة من مجموعتي البيانات لموازنة البيانات بالتساوي. سأقوم أيضًا ببعض عمليات تنظيف البيانات حتى نتمكن من التنبؤ بسهولة بالانتخابات الأمريكية:

```
reviews1 = trump_reviews[trump_reviews['Sentiment Polarity']
== 0.0000]
print(reviews1.shape)

cond1=trump_reviews['Sentiment
Polarity'].isin(reviews1['Sentiment Polarity'])
trump_reviews.drop(trump_reviews[cond1].index, inplace = True)
print(trump_reviews.shape)

reviews2 = biden_reviews[biden_reviews['Sentiment Polarity']
== 0.0000]
print(reviews2.shape)

cond2=biden_reviews['Sentiment
Polarity'].isin(reviews1['Sentiment Polarity'])
biden_reviews.drop(biden_reviews[cond2].index, inplace = True)
print(biden_reviews.shape)
```

الآن، قبل المضي قدمًا، نحتاج إلى موازنة مجموعتي البيانات:

```
# Donald Trump
np.random.seed(10)
remove_n =324
drop_indices = np.random.choice(trump_reviews.index, remove_n,
replace=False)
df_subset_trump = trump_reviews.drop(drop_indices)
print(df_subset_trump.shape)
# Joe Biden
np.random.seed(10)
remove_n =31
drop_indices = np.random.choice(biden_reviews.index, remove_n,
replace=False)
df_subset_biden = biden_reviews.drop(drop_indices)
print(df_subset_biden.shape)
```

الآن دعونا نحلل البيانات للتنبؤ بالانتخابات الأمريكية، من خلال تحليل عدد المشاعر الإيجابية والسلبية في كلا الحسابين:

```
count_1 = df_subset_trump.groupby('Expression Label').count()
print(count_1)

negative_per1 = (count_1['Sentiment Polarity'][0]/1000)*10
```

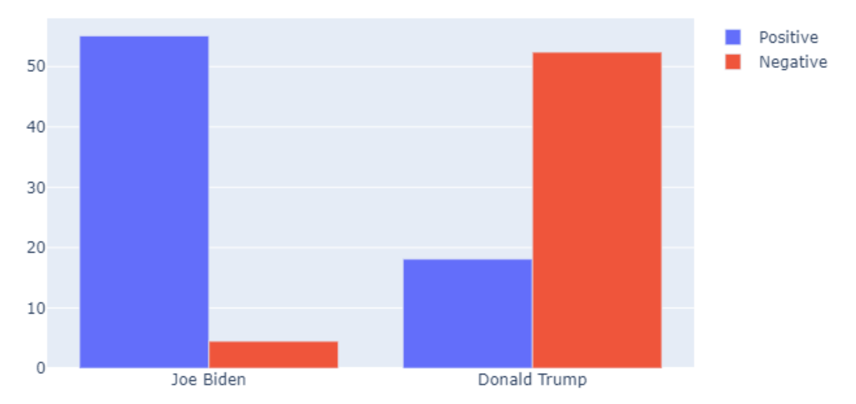
```
positive_per1 = (count_1['Sentiment Polarity'][1]/1000)*100

count_2 = df_subset_biden.groupby('Expression Label').count()
print(count_2)

negative_per2 = (count_2['Sentiment Polarity'][0]/1000)*100
positive_per2 = (count_2['Sentiment Polarity'][1]/1000)*100

Politicians = ['Joe Biden', 'Donald Trump']
lis_pos = [positive_per1, positive_per2]
lis_neg = [negative_per1, negative_per2]

fig = go.Figure(data=[
    go.Bar(name='Positive', x=Politicians, y=lis_pos),
    go.Bar(name='Negative', x=Politicians, y=lis_neg)
])
# Change the bar mode
fig.update_layout(barmode='group')
fig.show()
```



من الشكل أعلاه، من الواضح جداً أن جو بايدن يحصل على تغريدات أكثر إيجابية وتغريدات أقل سلبية مقارنةً بدونالد ترامب. لذلك لن يكون من الخطأ استنتاج أن جو بايدن يفضله الناس أكثر من دونالد ترامب للفوزفي الانتخابات الرئاسية الأمريكية.

أتمنى أن تكون قد أحببت هذا المقال حول كيفية توقع الفائزفي الانتخابات الأمريكية. يعتمد التحليل كليا على بيانات تويتر.

المصدر:

https://thecleverprogrammer.com/2020/10/01/predict-us-elections-with-python/

17) نموذج اكتشاف خطاب الكراهية Hate Speech Detection Model

يُفهم مصطلح خطاب الكراهية hate speech على أنه أي نوع من أنواع الاتصال اللفظي أو الكتابي أو السلوكي الذي يهاجم أو يستخدم لغة مهينة أو تمييزية ضد شخص أو مجموعة بناءً على ماهيتهم، وبعبارة أخرى، على أساس دينهم أو عرقهم أو جنسيتهم أو عرقهم. اللون أو النسب أو الجنس أو عامل هوية آخر.في هذه المقالة، سأطلعك على نموذج اكتشاف خطاب الكراهية باستخدام التعلم الآلي وبايثون.

يعد اكتشاف خطاب الكراهية عمومًا مهمة تصنيف المشاعر sentiment classification. لذلك بالنسبة للتدريب، يمكن تحقيق نموذج يمكنه تصنيف خطاب الكراهية من جزء معين من النص من خلال تدريبه على بيانات تُستخدم عمومًا لتصنيف المشاعر. لذلك بالنسبة لمهمة نموذج الكشف عن خطاب الكراهية، سأستخدم بيانات Twitter.

نموذج اكتشاف خطاب الكراهية

تتكون مجموعة البيانات التي سأستخدمها لنموذج الكشف عن الكلام الذي يحض على الكراهية من مجموعة اختبار وتدريب. تتضمن الحزمة التدريبية قائمة من 31962 تغريدة، ومعرف مطابق وعلامة 0 أو 1 لكل تغريدة. الشعور الخاص الذي نحتاج إلى اكتشافه في مجموعة البيانات هذه هو ما إذا كانت التغريدة تستند إلى خطاب الكراهية أم لا. يمكنك تنزيل مجموعة البيانات من هنا.

لذا، فلنبدأ بمهمة بناء نموذج لاكتشاف خطاب الكراهية. سأبدأ ببساطة بقراءة مجموعات البيانات باستخدام حزمة Pandas في لغة بايثون:

```
import pandas as pd
train = pd.read_csv('train.csv')
print("Training Set:"% train.columns, train.shape, len(train))
test = pd.read_csv('test.csv')
print("Test Set:"% test.columns, test.shape, len(test))
```

Training Set: (31962, 3) 31962
Test Set: (17197, 2) 17197

تنظيف البيانات

تنظيف البيانات Data cleaning هو عملية إعداد البيانات المنسقة بشكل غير صحيح للتحليل عن طريق حذف أو تعديل البيانات المنسقة بشكل غير صحيح والتي لا تكون ضرورية أو مفيدة بشكل عام لتحليل البيانات، حيث يمكن أن تعيق العملية أو تقدم نتائج غير دقيقة. الآن سأقوم بعملية تنظيف البيانات باستخدام مكتبة re في بايثون:

```
import re
def  clean_text(df, text_field):
    df[text_field] = df[text_field].str.lower()
    df[text_field] = df[text_field].apply(lambda elem:
re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z
\t])|(\w+:\/\/\S+)|^rt|http.+?", "", elem))
    return df
test_clean = clean_text(test, "tweet")
train_clean = clean_text(train, "tweet")
```

معالجة البيانات غير المتوازنة لنموذج اكتشاف خطاب الكراهية

إذا كنت ستحلل بعمق المهمة التي نعمل عليها مع سياق البيانات التي نستخدمها، فستجد أن التغريدات المتعلقة بخطابات الكراهية أقل نسبيًا من غيرها، لذا فهذه حالة من البيانات غير المتوازنة unbalanced data.

إذا كنا سنلائم هذه البيانات لتدريب نموذج الكشف عن خطاب الكراهية، فلن يعمم النموذج أي خطاب يحض على الكراهية لأن البيانات ذات السياق لخطاب الكراهية أقل بكثير من البيانات الإيجابية. لذلك في هذه الحالة، نحتاج إلى إعداد البيانات لتلائم نموذجنا بشكل صحيح.

هناك عدد من الطرق التي يمكنك استخدامها للتعامل مع هذا. يتمثل أحد الأساليب في استخدام إما الإفراط في أخذ العينات oversampling أو الاختزال downsampling. في حالة الإفراط في أخذ العينات، نستخدم دالة تقوم بشكل متكرر بأخذ عينات، مع الاستبدال، من فئة الأقلية minority class حتى يصبح الفئة بنفس حجم الأغلبية. دعونا نرى كيف يمكننا التعامل مع هذا:

```
from sklearn.utils import resample
train_majority = train_clean[train_clean.label==0]
train_minority = train_clean[train_clean.label==1]
train_minority_upsampled = resample(train_minority,
                                 replace=True,

n_samples=len(train_majority),
                                 random_state=123)
train_upsampled = pd.concat([train_minority_upsampled,
train_majority])
train_upsampled['label'].value_counts()
```

```
1 29720
0 29720
Name: label, dtype: int64
```

إنشاء المسار

لتبسيط نموذج اكتشاف خطاب الكراهية وإمكانية إعادة إنتاجه، سأستخدم مسار Scikit-Learn مع تصنيف SGD، قبل تدريب نموذجنا:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.linear_model import SGDClassifier
pipeline_sgd = Pipeline([
    ('vect', CountVectorizer()),
    ('tfidf',  TfidfTransformer()),
    ('nb', SGDClassifier()),])
```

تدريب نموذج اكتشاف خطاب الكراهية

الآن، قبل تدريب النموذج، دعنا نقسم البيانات إلى مجموعة تدريب ومجموعة اختبار:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(train_upsampled['tweet'],

train_upsampled['label'],random_state = 0)
```

الآن دعنا ندرب النموذج ونتوقع النتائج على مجموعة الاختبار باستخدام طريقة F1 score:

```
model = pipeline_sgd.fit(X_train, y_train)
y_predict = model.predict(X_test)
from sklearn.metrics import f1_score
f1_score(y_test, y_predict)
```

0.9696

لذلك حصلنا على درجة F1 بنسبة 0.96في المائة وهو أمر يستحق التقدير بشكل عام. يمكن الآن نشر هذا النموذج واستخدامهفي الإنتاج. أتمنى أن تكون قد أحببت هذه المقالة حول نموذج اكتشاف خطاب الكراهية باستخدام التعلم الآلي. لا تترددفي طرح أسئلتك القيمةفي قسم التعليقات أدناه.

المصدر:

https://thecleverprogrammer.com/2020/08/19/hate-speech-detection-/model

18) تحليل المشاعر على تويتر Twitter Sentiment Analysis

تحليل المشاعر على Twitter هو عملية تحديد وتصنيف التغريدات بشكل حسابي في جزء من النص، لا سيما من أجل تحديد ما إذا كان موقف الكاتب تجاه موضوع أو منتج معين، وما إلى ذلك، إيجابيًا أم سلبيًا أم محايدًا.

في هذه المقالة سأقوم بتحليل المشاعر على تويتر باستخدام المعالجة اللغوية الطبيعية Natural Language Processing باستخدام مكتبة nltk مع بايثون.

تحليل المشاعر على تويتر

استيراد المكتبات:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)
from sklearn.model_selection import train_test_split #
function for splitting data to train and test sets

import nltk
from nltk.corpus import stopwords
from nltk.classify import SklearnClassifier

from wordcloud import WordCloud,STOPWORDS
import matplotlib.pyplot as plt
```

تنزيل مجموعة البيانات:

مجموعة البيانات.

```
data = pd.read_csv('Sentiment.csv')
 #Keeping only the neccessary columns
data = data[['text','sentiment']]
```

بادئ ذي بدء، تقسيم مجموعة البيانات إلى مجموعة تدريب واختبار. مجموعة الاختبار هي 10٪ من مجموعة البيانات الأصلية.

بالنسبة لهذا التحليل الخاص، أسقطت التغريدات المحايدة، حيث كان هدفي هو التمييز بين التغريدات الإيجابية والسلبية فقط.

```
 #Splitting the dataset into train and test set
train, test = train_test_split(data,test_size = 0.1)
 #Removing neutral sentiments
train = train[train.sentiment != "Neutral"]
```

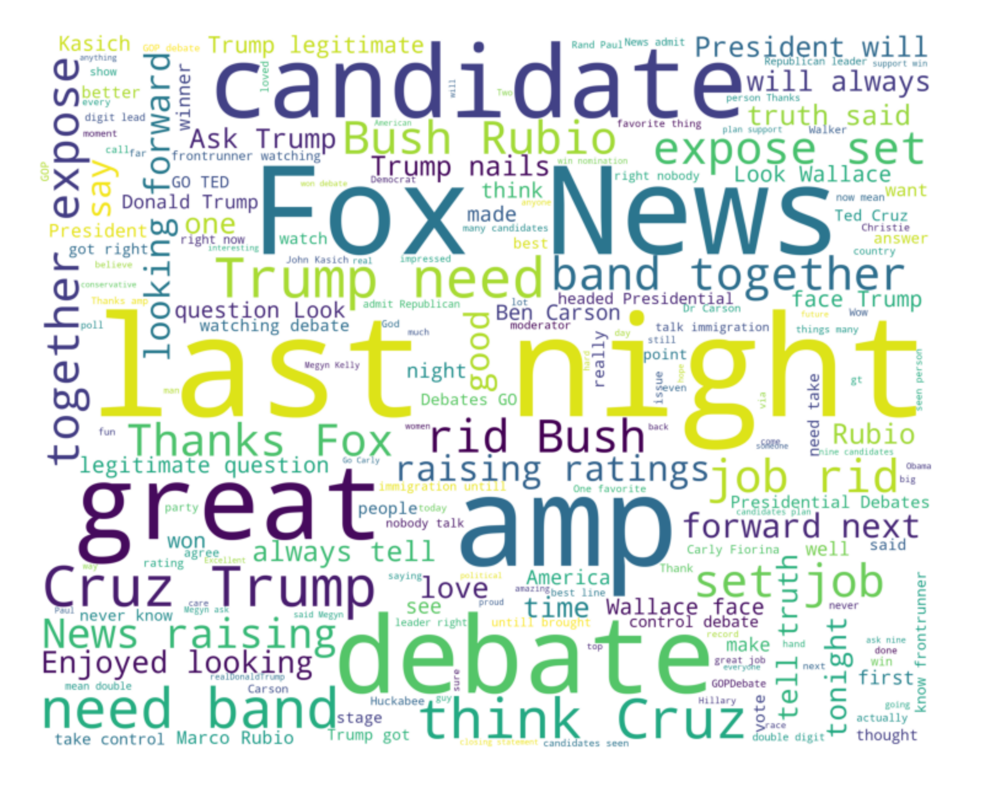
كخطوة تالية، قمت بفصل التغريدات الإيجابية والسلبية لمجموعة التدريب من أجل تصوير الكلمات المضمنة بسهولة.

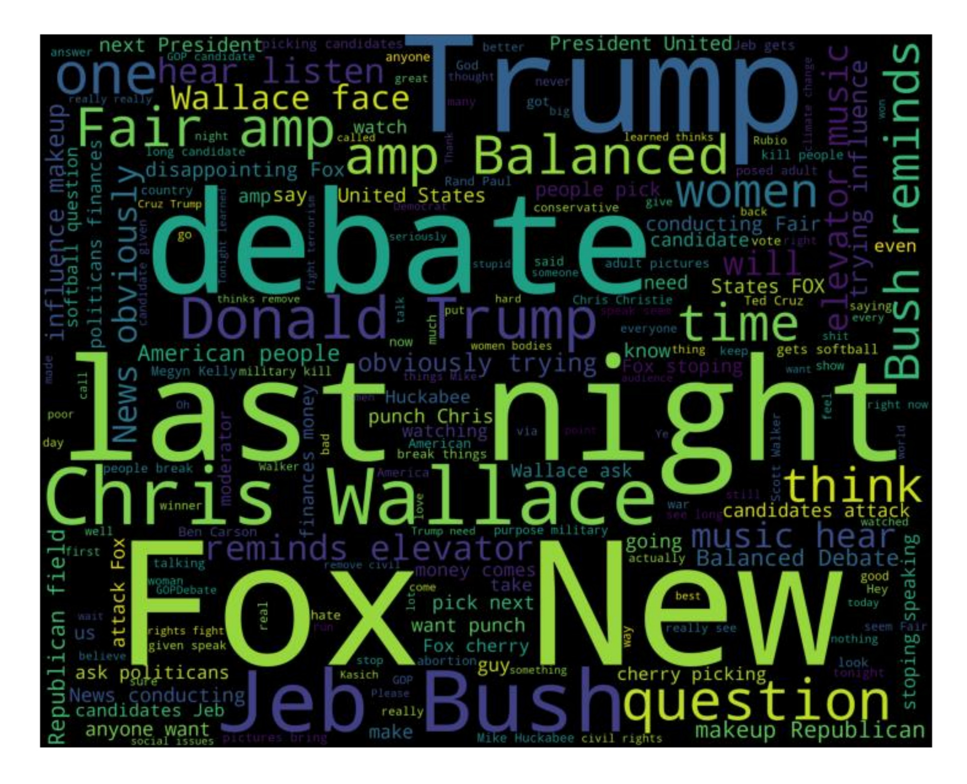
بعد ذلك قمت بتنظيف النص من علامات التصنيف والإشارات والروابط. أصبحوا الآن جاهزين لتصوير WordCloud الذي يعرض فقط الكلمات الأكثر تأكيدًا للتغريدات الإيجابية والسلبية.

```
train_pos = train[ train['sentiment'] == 'Positive']
train_pos = train_pos['text']
train_neg = train[ train['sentiment'] == 'Negative']
train_neg = train_neg['text']

def wordcloud_draw(data, color = 'black'):
    words = ' '.join(data)
    cleaned_word = " ".join([word for word in words.split()
                            if 'http' not in word
                                and not word.startswith('@')
                                and not word.startswith('#')
                                and word != 'RT'
([
    wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color=color,
                      width=2500,
                      height=2000
.(                     generate(cleaned_word(
    plt.figure(1,figsize=(13, 13))
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.show()

print("Positive words")
wordcloud_draw(train_pos,'white')
print("Negative words")
wordcloud_draw(train_neg)
```





من المثير للاهتمام ملاحظة الكلمات والعبارات التاليةفي مجموعة الكلمات الإيجابية: truth، strong، legitimate، together، love، job.

في تفسيري، يميل الناس إلى الاعتقاد بأن مرشحهم المثالي صادق وشرعي وفوق الخير والشر.

في الوقت نفسه، تحتوي التغريدات السلبية على كلمات مثل: influence، news، Elevatormusic ، disappointing، softball، makeup، cherry picking، trying.

حسب فهمي، فات الناس التمثيل الحاسم واعتبروا المرشحين الموبخين ضعيفين للغاية والتقاطيون.

بعد التصوير، قمت بإزالة علامات التجزئة والإشارات والروابط وكلمات التوقف من مجموعة التدريب.

كلمات التوقفStop Words : هي الكلمات التي لا تحتوي على أهمية مهمة لاستخدامهافي استعلامات البحث.

عادةً ما يتم تصفية هذه الكلمات من استعلامات البحث لأنها تُرجع قدرًا هائلاً من المعلومات غير الضرورية. (the، for، this، إلخ.)

```
tweets =[]
stopwords_set = set(stopwords.words("english"))

for index, row in train.iterrows:()
    words_filtered = [e.lower() for e in row.text.split() if
len(e) >= 3]
    words_cleaned = [word for word in words_filtered
        if 'http' not in word
        and not word.startswith('@')
        and not word.startswith('#')
        and word != 'RT['
    words_without_stopwords = [word for word in words_cleaned
if not word in stopwords_set]
    tweets.append((words_without_stopwords, row.sentiment))

test_pos = test[ test['sentiment'] == 'Positive']
test_pos = test_pos['text']
test_neg = test[ test['sentiment'] == 'Negative']
test_neg = test_neg['text']
```

كخطوة تالية، قمت باستخراج الميزات المسماة باستخدام nltk lib ، أولاً عن طريق قياس التوزيع المتكرر واختيار المفاتيح الناتجة.

```
#Extracting word features
def get_words_in_tweets(tweets):
  all[] =
  for (words, sentiment) in tweets:
    all.extend(words)
  return all

def get_word_features(wordlist):
  wordlist = nltk.FreqDist(wordlist)
  features = wordlist.keys()
  return features
w_features = get_word_features(get_words_in_tweets(tweets))

def extract_features(document):
  document_words = set(document)
  features{} =
  for word in w_features:
    features['contains(%s)' % word] = (word in document_words)
    return features
```

بموجب هذا قمت برسم الكلمات الأكثر انتشارًا. تتركز معظم الكلمات حول ليالي النقاش .debate nights

```
wordcloud_draw(w_features)
```

باستخدام Classifier NaiveBayes nltk، قمت بتصنيف ميزات كلمات التغريدة المستخرجة.

```
#Training the Naive Bayes classifier
```

```
training_set =
nltk.classify.apply_features(extract_features,tweets)
classifier = nltk.NaiveBayesClassifier.train(training_set)
```

أخيرًا، باستخدام مقاييس غير ذكية، حاولت قياس كيفية تسجيل خوارزمية المصنف.

```
neg_cnt = 0
pos_cnt = 0
for obj in test_neg:
  res =  classifier.classify(extract_features(obj.split()))
  if(res == 'Negative'):
    neg_cnt = neg_cnt + 1
for obj in test_pos:
  res =  classifier.classify(extract_features(obj.split()))
  if(res == 'Positive'):
    pos_cnt = pos_cnt + 1

print('[Negative]: %s/%s '  % (len(test_neg),neg_cnt))
print('[Positive]: %s/%s '  % (len(test_pos),pos_cnt))
```

```
[Negative]: 842/795
[Positive]: 220/74
```

المصدر:

https://thecleverprogrammer.com/2020/06/13/twitter-sentiment-analysis/

19) تحليل مراجعات الفيلم Movie Reviews Sentiment Analysis

في مشروع التعلم الآلي هذا، سننشئ تصنيفًا ثنائيًا يضع نصوص مراجعات الأفلامفي إحدى فئتين_ المشاعر السلبية أو الإيجابية. سنلقي نظرة سريعة على نظرية بايز Bayes theorem ونخفف متطلباتها باستخدام افتراض نايف Naive assumption.

لنبدأ باستيراد المكتبات:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)
import re # for regex
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import
GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score
import pickle
```

يمكنك تنزيل مجموعة البيانات التي تحتاجها لهذه المهمة من هنا:

IMDB-Dataset   Download DataSEt

```
data = pd.read_csv('IMDB Dataset.csv')
print(data.shape)
data.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

```
data.info()
```

```
#Output
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   review     50000 non-null  object
 1   sentiment  50000 non-null  object
dtypes: object(2)
memory usage: 781.4+ KB
```

لا توجد قيم فارغة، قم بترميز تسمية Label encode المشاعر إلى 1 (موجب) و 0 (سلبي):

```
data.sentiment.value_counts()
```

```
#Output
positive    25000
negative    25000
Name: sentiment, dtype: int64
```

```
data.sentiment.replace('positive',1,inplace=True)
data.sentiment.replace('negative',0,inplace=True)
data.head(10)
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	1
1	A wonderful little production. The...	1
2	I thought this was a wonderful way to spend ti...	1
3	Basically there's a family where a little boy ...	0
4	Petter Mattei's "Love in the Time of Money" is...	1
5	Probably my all-time favorite movie, a story o...	1
6	I sure would like to see a resurrection of a u...	1
7	This show was an amazing, fresh & innovative i...	0
8	Encouraged by the positive comments about this...	0
9	If you like original gut wrenching laughter yo...	1

```
data.review[0]
```

```
#Output
One of the other reviewers has mentioned that after watching
just 1 Oz episode you'll be hooked. They are right, as this is
exactly what happened with me.<br /><br />The first thing that
struck me about Oz was its brutality and unflinching scenes of
violence, which set in right from the word GO. Trust me, this
is not a show for the faint hearted or timid. This show pulls
no punches with regards to drugs, sex or violence. Its is
hardcore, in the classic use of the word.<br /><br />It is
called OZ as that is the nickname given to the Oswald Maximum
Security State Penitentary. It focuses mainly on Emerald City,
an experimental section of the prison where all the cells have
glass fronts and face inwards, so privacy is not high on the
agenda. Em City is home to many..Aryans, Muslims, gangstas,
Latinos, Christians, Italians, Irish and more....so scuffles,
death stares, dodgy dealings and shady agreements are never
far away.<br /><br />I would say the main appeal of the show
is due to the fact that it goes where other shows wouldn't
dare. Forget pretty pictures painted for mainstream audiences,
forget charm, forget romance...OZ doesn't mess around. The
first episode I ever saw struck me as so nasty it was surreal,
I couldn't say I was ready for it, but as I watched more, I
developed a taste for Oz, and got accustomed to the high
levels of graphic violence. Not just violence, but injustice
(crooked guards who'll be sold out for a nickel, inmates
who'll kill on order and get away with it, well mannered,
middle class inmates being turned into prison bitches due to
their lack of street skills or prison experience) Watching Oz,
you may become comfortable with what is uncomfortable
viewing....thats if you can get in touch with your darker side.
```

خطوات لتنظيف المراجعات:

1. إزالة علامات HTML Remove HTML tags.
2. إزالة الأحرف الخاصة Remove special characters.
3. تحويل كل شيء إلى أحرف صغيرة Convert everything to lowercase.
4. إزالة كلمات التوقف Remove stopwords.
5. التجذيع Stemming.

1. إزالة علامات HTML

```
def clean(text):
    cleaned = re.compile(r'<.*?>')
    return re.sub(cleaned,'',text)

data.review = data.review.apply(clean)
data.review[0]
```

```
#Output
"One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me.The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or timid. This show pulls no punches with regards to drugs, sex or violence. Its is hardcore, in the classic use of the word.It is called OZ as that is the nickname given to the Oswald Maximum Security State Penitentary. It focuses mainly on Emerald City, an experimental section of the prison where all the cells have glass fronts and face inwards, so privacy is not high on the agenda. Em City is home to many..Aryans, Muslims, gangstas, Latinos, Christians, Italians, Irish and more....so scuffles, death stares, dodgy dealings and shady agreements are never far away.I would say the main appeal of the show is due to the fact that it goes where other shows wouldn't dare. Forget pretty pictures painted for mainstream audiences, forget charm, forget romance...OZ doesn't mess around. The first episode I ever saw struck me as so nasty it was surreal, I couldn't say I was ready for it, but as I watched more, I developed a taste for Oz, and got accustomed to the high levels of graphic violence. Not just violence, but injustice (crooked guards who'll be sold out for a nickel, inmates who'll kill on order and get away with it, well mannered, middle class inmates being turned into prison bitches due to their lack of street skills or prison experience) Watching Oz, you may become comfortable with what is uncomfortable viewing....thats if you can get in touch with your darker side".
```

2. إزالة الأحرف الخاصة

```
def is_special(text):
    rem'' =
    for i in text:
        if i.isalnum:()
            rem = rem + i
        else:
            rem = rem' ' +
    return rem

data.review = data.review.apply(is_special)
data.review[0]
```

```
#Output
'One of the other reviewers has mentioned that after watching just 1 Oz episode you ll be hooked  They are right  as this is exactly what happened with me The first thing that struck me about Oz was its brutality and unflinching scenes of violence which set in right from the word GO  Trust me  this is not a show for the faint hearted or timid  This show pulls no punches with regards to drugs  sex or violence  Its is hardcore  in the classic use of the word It is called OZ as that is the nickname given to the Oswald Maximum Security State Penitentary  It focuses mainly on Emerald City  an experimental section of the
```

```
prison where all the cells have glass fronts and face inwards
so privacy is not high on the agenda  Em City is home to many
Aryans  Muslims  gangstas  Latinos  Christians  Italians  Irish
and more    so scuffles  death stares  dodgy dealings and shady
agreements are never far away I would say the main appeal of the
show is due to the fact that it goes where other shows wouldn t
dare  Forget pretty pictures painted for mainstream audiences
forget charm  forget romance   OZ doesn t mess around  The first
episode I ever saw struck me as so nasty it was surreal  I couldn
t say I was ready for it  but as I watched more  I developed a
taste for Oz  and got accustomed to the high levels of graphic
violence  Not just violence  but injustice  crooked guards who
ll be sold out for a nickel  inmates who ll kill on order and
get away with it   well mannered  middle class inmates being
turned into prison bitches due to their lack of street skills or
prison experience  Watching Oz  you may become comfortable with
what is uncomfortable viewing    thats if you can get in touch
with your darker side '
```

3. تحويل كل شيء إلى أحرف صغيرة

```
def to_lower(text):
    return text.lower()

data.review = data.review.apply(to_lower)
data.review[0]
```

```
#Output
'one of the other reviewers has mentioned that after watching
just 1 oz episode you ll be hooked  they are right  as this is
exactly what happened with me the first thing that struck me
about oz was its brutality and unflinching scenes of violence
which set in right from the word go  trust me  this is not a
show for the faint hearted or timid  this show pulls no punches
with regards to drugs  sex or violence  its is hardcore  in the
classic use of the word it is called oz as that is the nickname
given to the oswald maximum security state penitentary  it
focuses mainly on emerald city  an experimental section of the
prison where all the cells have glass fronts and face inwards
so privacy is not high on the agenda  em city is home to many
aryans  muslims  gangstas  latinos  christians  italians  irish
and more    so scuffles  death stares  dodgy dealings and shady
agreements are never far away i would say the main appeal of the
show is due to the fact that it goes where other shows wouldn t
dare  forget pretty pictures painted for mainstream audiences
forget charm  forget romance   oz doesn t mess around  the first
episode i ever saw struck me as so nasty it was surreal  i couldn
t say i was ready for it  but as i watched more  i developed a
taste for oz  and got accustomed to the high levels of graphic
violence  not just violence  but injustice  crooked guards who
ll be sold out for a nickel  inmates who ll kill on order and
get away with it   well mannered  middle class inmates being
turned into prison bitches due to their lack of street skills or
prison experience  watching oz  you may become comfortable with
```

```
what is uncomfortable viewing    thats if you can get in touch
with your darker side'
```

4. إزالة كلمات التوقف

```
def rem_stopwords(text):
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(text)
    return [w for w in words if w not in stop_words]

data.review = data.review.apply(rem_stopwords)
data.review[0]
```

5. جذع الكلمات

```
def stem_txt(text):
    ss = SnowballStemmer('english')
    return " ".join([ss.stem(w) for w in text])

data.review = data.review.apply(stem_txt)
data.review[0]
```

```
data.head()
```

	review	sentiment
0	one review mention watch 1 oz episod hook righ...	1
1	wonder littl product film techniqu unassum old...	1
2	thought wonder way spend time hot summer weeke...	1
3	basic famili littl boy jake think zombi closet...	0
4	petter mattei love time money visual stun film...	1

إنشاء النموذج

1. إنشاء حقيبة الكلمات Bag Of Words (BOW)

```
X = np.array(data.iloc[:,0].values)
y = np.array(data.sentiment.values)
cv = CountVectorizer(max_features = 1000)
X = cv.fit_transform(data.review).toarray()
print("X.shape = ",X.shape)
print("y.shape = ",y.shape)
```

```
#Output
X.shape =  (50000, 1000)
y.shape =  (50000,)
```

```
print(X)
```

```
#Output
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 1, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 1, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

2. تقسيم تدريب/اختبار

```
trainx,testx,trainy,testy =
train_test_split(X,y,test_size=0.2,random_state=9)
print("Train shapes : X = {}, y =
{}".format(trainx.shape,trainy.shape))
print("Test shapes : X = {}, y =
{}".format(testx.shape,testy.shape))
```

```
#Output
Train shapes : X = (40000, 1000), y = (40000,)
Test shapes : X = (10000, 1000), y = (10000,)
```

3. تعريف النماذج والتدريب عليها

```
#Output
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None,
fit_prior=True)
```

4. مقاييس التنبؤ والدقة لاختيار النموذج الأفضل

```
ypg = gnb.predict(testx)
ypm = mnb.predict(testx)
ypb = bnb.predict(testx)

print("Gaussian = ",accuracy_score(testy,ypg))
print("Multinomial = ",accuracy_score(testy,ypm))
print("Bernoulli = ",accuracy_score(testy,ypb))
```

```
#Output
Gaussian =  0.7843
Multinomial =  0.831
Bernoulli =  0.8386
```

```
pickle.dump(bnb,open('model1.pkl','wb'))
```

```
rev =  """Terrible. Complete trash. Brainless tripe. Insulting
to anyone who isn't an 8 year old fan boy. Im actually pretty
```

```
disgusted that this movie is making the money it is - what does
it say about the people who brainlessly hand over the hard earned
cash to be 'entertained' in this fashion and then come here to
leave a positive 8.8 review?? Oh yes, they are morons. Its the
only sensible conclusion to draw. How anyone can rate this movie
amongst the pantheon of great titles is beyond me.

So trying to find something constructive to say about this title
is hard...I enjoyed Iron Man? Tony Stark is an inspirational
character in his own movies but here he is a pale shadow of
that...About the only 'hook' this movie had into me was wondering
when and if Iron Man would knock Captain America out...Oh how I
wished he had :( What were these other characters anyways?
Useless, bickering idiots who really couldn't organise happy
times in a brewery. The film was a chaotic mish mash of action
elements and failed 'set pieces...'

I found the villain to be quite amusing.

And now I give up. This movie is not robbing any more of my time
but I felt I ought to contribute to restoring the obvious fake
rating and reviews this movie has been getting on IMDb""".
f1 = clean(rev)
f2 = is_special(f1)
f3 = to_lower(f2)
f4 = rem_stopwords(f3)
f5 = stem_txt(f4)
```

```
bow,words = [],word_tokenize(f5)
for word in words:
    bow.append(words.count(word))
#np.array(bow).reshape(1,3000)
#bow.shape
word_dict = cv.vocabulary_
pickle.dump(word_dict,open('bow.pkl','wb'))
```

[0]

0 يعني سلبي.

المصدر:

https://thecleverprogrammer.com/2020/05/25/movie-reviews-sentiment-analysis-binary-classification-with-machine-learning/

20) تحليل آراء مراجعات منتجات أمازون باستخدام التعلم الآلي Amazon Product Reviews Sentiment Analysis with Machine Learning

أصبحت مراجعات المنتجات Product reviews أكثر أهمية مع تطور متاجر البيع بالتجزئة التقليدية من الطوب والملاط إلى التسوق عبر الإنترنت.

ينشر المستهلكون مراجعاتهم مباشرة على صفحات المنتجفي الوقت الفعلي. مع الكم الهائل من آراء المستهلكين، فإن هذا يخلق فرصة لمعرفة كيف يتفاعل السوق مع منتج معين.

سنحاول معرفة ما إذا كان بإمكاننا التنبؤ بمشاعر مراجعة المنتج باستخدام بايثون والتعلم الآلي.

دعنا نستورد الوحدات الضرورية ونلقي نظرة على البيانات:

يمكنك تنزيل مجموعة البيانات هذه من هنا.

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import math
import warnings
warnings.filterwarnings('ignore') # Hides warning
warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.filterwarnings("ignore",category=UserWarning)
sns.set_style("whitegrid") # Plotting style
np.random.seed(7) # seeding random number generator

df = pd.read_csv('amazon.csv')
print(df.head())
```

```
                     id  ... reviews.username
0  AVqkIhwDv8e3D1O-lebb  ...          Adapter
1  AVqkIhwDv8e3D1O-lebb  ...           truman
2  AVqkIhwDv8e3D1O-lebb  ...            DaveZ
3  AVqkIhwDv8e3D1O-lebb  ...           Shacks
4  AVqkIhwDv8e3D1O-lebb  ...        explore42

[5 rows x 21 columns]
```

وصف مجموعة البيانات

```
data = df.copy()
data.describe()
```

	reviews.id	reviews.numHelpful	reviews.rating	reviews.userCity	reviews.userProvince
count	1.0	34131.000000	34627.000000	0.0	0.0
mean	111372787.0	0.630248	4.584573	NaN	NaN
std	NaN	13.215775	0.735653	NaN	NaN
min	111372787.0	0.000000	1.000000	NaN	NaN
25%	111372787.0	0.000000	4.000000	NaN	NaN
50%	111372787.0	0.000000	5.000000	NaN	NaN
75%	111372787.0	0.000000	5.000000	NaN	NaN
max	111372787.0	814.000000	5.000000	NaN	NaN

```
data.info ()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34660 entries, 0 to 34659
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    34660 non-null  object
 1   name                  27900 non-null  object
 2   asins                 34658 non-null  object
 3   brand                 34660 non-null  object
 4   categories            34660 non-null  object
 5   keys                  34660 non-null  object
 6   manufacturer          34660 non-null  object
 7   reviews.date          34621 non-null  object
 8   reviews.dateAdded     24039 non-null  object
 9   reviews.dateSeen      34660 non-null  object
 10  reviews.didPurchase   1 non-null      object
 11  reviews.doRecommend   34066 non-null  object
 12  reviews.id            1 non-null      float64
 13  reviews.numHelpful    34131 non-null  float64
 14  reviews.rating        34627 non-null  float64
 15  reviews.sourceURLs    34660 non-null  object
 16  reviews.text          34659 non-null  object
 17  reviews.title         34655 non-null  object
 18  reviews.userCity      0 non-null      float64
 19  reviews.userProvince  0 non-null      float64
 20  reviews.username      34658 non-null  object
dtypes: float64(5), object(16)
memory usage: 5.6+ MB
```

نحتاج إلى تنظيف عمود الاسم name column من خلال الرجوع إلى asins (منتجات فريدة unique products) نظرًا لأن لدينا 7000 قيمة مفقودة:

```
data["asins"].unique()
```

```
array(['B01AHB9CN2', 'B00VINDBJK', 'B005PB2T0S', 'B002Y27P3M',
       'B01AHB9CYG', 'B01AHB9C1E', 'B01J2G4VBG', 'B00ZV9PXP2',
       'B0083Q04TA', 'B018Y229OU', 'B00REQKWGA', 'B00IOYAM4I',
       'B018T075DC', nan, 'B00DU15MU4', 'B018Y225IA', 'B005PB2T2Q',
       'B018Y23MNM', 'B00OQVZDJM', 'B00IOY8XWQ', 'B00LO29KXQ',
       'B00QJDU3KY', 'B018Y22C2Y', 'B01BFIBRIE', 'B01J4ORNHU',
       'B018SZT3BK', 'B00UH4D8G2', 'B018Y22BI4', 'B00TSUGXKE',
       'B00L9EPT8O,B01E6AO69U', 'B018Y23P7K', 'B00X4WHP5E', 'B00QFQRELG',
       'B00LW9XOJM', 'B00QL1ZN3G', 'B0189XYY0Q', 'B01BH83OOM',
       'B00BFJAHF8', 'B00U3FPN4U', 'B002Y27P6Y', 'B006GWO5NE',
       'B006GWO5WK'], dtype=object)
```

```
asins_unique = len(data["asins"].unique())
print("Number of Unique ASINs: " + str(asins_unique))
```

#Output– Number of Unique ASINs: 42

تصوير توزيعات المتغيرات العددية:

```
data.hist(bins=50, figsize=(20,15))
plt.show()
```

reviews.id

reviews.numHelpful

reviews.rating

reviews.userCity

reviews.userProvince

تعتبر القيم المتطرفة Outliers في هذه الحالة ذات قيمة، لذلك قد نرغب في تقييم التقييمات التي تضم أكثر من 50 شخصًا يجدونها مفيدة.

تم تصنيف غالبية الأمثلة بدرجة عالية (بالنظر إلى توزيع التصنيف). هناك ضعف مقدار تصنيفات 5 نجوم من التصنيفات الأخرى مجتمعة.

تقسيم البيانات إلى تدريب واختبار

قبل أن نستكشف مجموعة البيانات، سنقسمها إلى مجموعة تدريب ومجموعات اختبار. هدفنا في النهاية هو تدريب مصنف تحليل المشاعر.

نظرًا لأن غالبية المراجعات إيجابية (5 نجوم)، فسنحتاج إلى إجراء تقسيم طبقي stratified split على درجة المراجعات reviews score لضمان عدم تدريب المصنف على البيانات غير المتوازنة imbalanced data.

```
from sklearn.model_selection import StratifiedShuffleSplit
print("Before {}".format(len(data)))
dataAfter = data.dropna(subset=["reviews.rating"])
 #Removes all NAN in reviews.rating
print("After {}".format(len(dataAfter)))
dataAfter["reviews.rating"]                                      =
dataAfter["reviews.rating"].astype(int)

split = StratifiedShuffleSplit(n_splits=5, test_size=0.2)
for train_index, test_index in split.split(dataAfter,

dataAfter["reviews.rating"]):
    strat_train = dataAfter.reindex(train_index)
    strat_test = dataAfter.reindex(test_index)
```

Before 34660

After 34627

نحتاج إلى معرفة ما إذا كانت مجموعات التدريب والاختبار مقسمة بشكل متناسب مقارنة بالبيانات الأولية:

```
print(len(strat_train))
print(len(strat_test))
print(strat_test["reviews.rating"].value_counts()/len(strat_te
st))
```

```
27701
6926
5.0    0.689864
4.0    0.244730
3.0    0.042160
1.0    0.011406
2.0    0.011118
Name: reviews.rating, dtype: float64
```

استكشاف البيانات (مجموعة التدريب)

سنستخدم التعابير المنتظمة regular expressions لتنظيف أي أحرف غير مرغوب بهافي مجموعة البيانات، ثم معاينة الشكل الذي تبدو عليه البيانات بعد التنظيف.

```
reviews = strat_train.copy()
reviews.head()
```

```
                          id  ... Sentiment
4349   AVphgVaX1cnluZ0-DR74  ...  Positive
30776  AV1YE_muvKc47QAVgpwE  ...  Positive
28775  AVpidLjVilAPnD_xEVpI  ...   Neutral
1136   AVqkIhwDv8e3D1O-lebb  ...  Positive
17803  AV1YnRtnglJLPUi8IJmV  ...  Positive

[5 rows x 22 columns]
```

```
print(len(reviews["name"].unique()),
len(reviews["asins"].unique()))
print(reviews.info())
print(reviews.groupby("asins")["name"].unique())
```

```
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    27673 non-null  object
 1   name                  22268 non-null  object
 2   asins                 27672 non-null  object
 3   brand                 27673 non-null  object
 4   categories            27673 non-null  object
 5   keys                  27673 non-null  object
 6   manufacturer          27673 non-null  object
 7   reviews.date          27659 non-null  object
 8   reviews.dateAdded     19228 non-null  object
 9   reviews.dateSeen      27673 non-null  object
 10  reviews.didPurchase   1 non-null      object
 11  reviews.doRecommend   27257 non-null  object
 12  reviews.id            1 non-null      float64
 13  reviews.numHelpful    27304 non-null  float64
 14  reviews.rating        27673 non-null  float64
 15  reviews.sourceURLs    27673 non-null  object
 16  reviews.text          27672 non-null  object
 17  reviews.title         27668 non-null  object
 18  reviews.userCity      0 non-null      float64
 19  reviews.userProvince  0 non-null      float64
 20  reviews.username      27671 non-null  object
dtypes: float64(5), object(16)
memory usage: 4.6+ MB
```

```
#Output
Echo (White),,,
Echo (White),,,
Amazon Fire Tv,,,
Amazon Fire Tv,,,
nan
Amazon - Amazon Tap Portable Bluetooth and Wi-Fi Speaker - Black,,,
Amazon - Amazon Tap Portable Bluetooth and Wi-Fi Speaker - Black,,,
Amazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special Offers - Silver Aluminum,,,
Amazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special Offers - Silver Aluminum,,,
Amazon 9W PowerFast Official OEM USB Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,
```

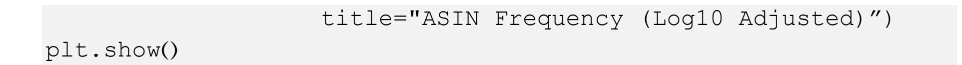
```
Amazon 9W PowerFast Official OEM USB Charger and Power Adapter
for Fire Tablets and Kindle eReaders,,,
Amazon Kindle Fire 5ft USB to Micro-USB Cable (works with most
Micro-USB Tablets),,,
Amazon Kindle Fire 5ft USB to Micro-USB Cable (works with most
Micro-USB Tablets),,,
Kindle Dx Leather Cover, Black (fits 9.7 Display, Latest and 2nd
Generation Kindle Dxs),,
Amazon Fire Hd 6 Standing Protective Case(4th Generation - 2014
Release), Cayenne Red,,,
Amazon Fire Hd 6 Standing Protective Case(4th Generation - 2014
Release), Cayenne Red,,,
Amazon Fire Hd 6 Standing Protective Case(4th Generation - 2014
Release), Cayenne Red,,,
Amazon 5W USB Official OEM Charger and Power Adapter for Fire
Tablets and Kindle eReaders,,,
New Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro
Usb Angle Cable,,,
New Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro
Usb Angle Cable,,,
Amazon 5W USB Official OEM Charger and Power Adapter for Fire
Tablets and Kindle eReaders,,,
Amazon 5W USB Official OEM Charger and Power Adapter for Fire
Tablets and Kindle eReaders,,,
Echo (White),,,
Fire Tablet, 7 Display, Wi-Fi, 8 GB - Includes Special Offers,
Tangerine"
Echo (Black),,,
Amazon 9W PowerFast Official OEM USB Charger and Power Adapter
for Fire Tablets and Kindle eReaders,,,
Echo (Black),,,
Echo (Black),,,
Amazon Fire Tv,,,
Kindle Dx Leather Cover, Black (fits 9.7 Display, Latest and 2nd
Generation Kindle Dxs),,"
New Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro
Usb Angle Cable,,,

Echo                (White),,,\r\nEcho                (White),,,
2318
Amazon       Fire        Tv,,,\r\nAmazon       Fire        Tv,,,
2029
Amazon - Amazon Tap Portable Bluetooth and Wi-Fi Speaker -
Black,,,\r\nAmazon - Amazon Tap Portable Bluetooth and Wi-Fi
Speaker                        -                       Black,,,
259
Amazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special Offers - Silver
Aluminum,,,\r\nAmazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special
```

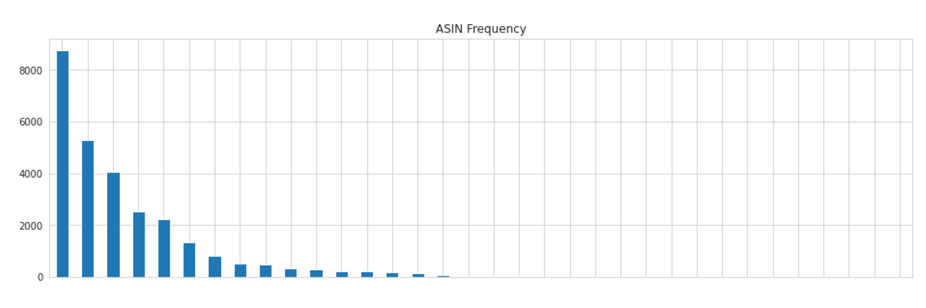
```
Offers - Silver Aluminum,,,
106
Amazon 9W PowerFast Official OEM USB Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,\r\nAmazon 9W PowerFast Official OEM USB Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,      28
Kindle Dx Leather Cover, Black (fits 9.7 Display, Latest and 2nd Generation Kindle Dxs),,
7
Amazon 5W USB Official OEM Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,\r\nAmazon 5W USB Official OEM Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,
5
Amazon Fire Hd 6 Standing Protective Case(4th Generation - 2014 Release), Cayenne Red,,,\r\nAmazon Fire Hd 6 Standing Protective Case(4th Generation - 2014 Release), Cayenne Red,,,
5
New Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro Usb Angle Cable,,,\r\nNew Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro Usb Angle Cable,,,
5
Amazon Kindle Fire 5ft USB to Micro-USB Cable (works with most Micro-USB Tablets),,,\r\nAmazon Kindle Fire 5ft USB to Micro-USB Cable (works with most Micro-USB Tablets),,,
4
Echo (Black),,,\r\nEcho (Black),,,
3
Echo (White),,,\r\nFire Tablet, 7 Display, Wi-Fi, 8 GB - Includes Special Offers, Tangerine"
1
Amazon Fire Hd 6 Standing Protective Case(4th Generation - 2014 Release), Cayenne Red,,,\r\nAmazon 5W USB Official OEM Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,
1
Echo (Black),,,\r\nAmazon 9W PowerFast Official OEM USB Charger and Power Adapter for Fire Tablets and Kindle eReaders,,,
1
New Amazon Kindle Fire Hd 9w Powerfast Adapter Charger + Micro Usb Angle Cable,,,\r\n
1
Amazon Fire Tv,,,\r\nKindle Dx Leather Cover, Black (fits 9.7 Display, Latest and 2nd Generation Kindle Dxs)",,
1
Name: name, dtype: int64
```

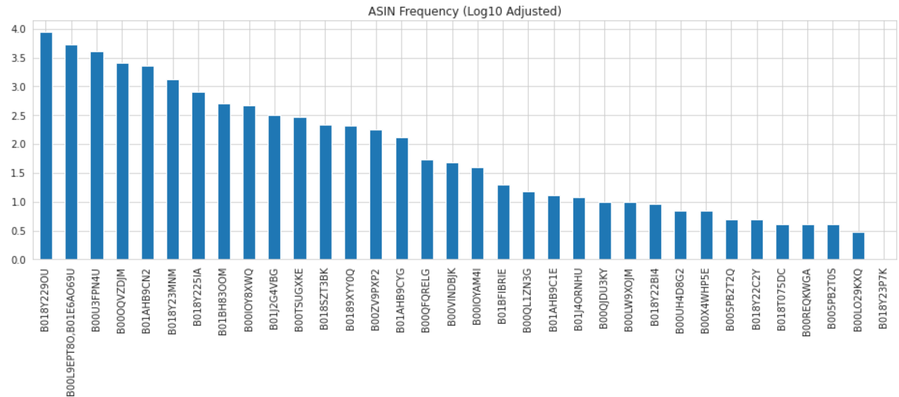
أكد الناتج أن كل ASIN يمكن أن يكون له أسماء متعددة. لذلك يجب علينا فقط أن نهتم بأنفسنا بشأن ما تفعله ASIN بشكل جيد، وليس أسماء المنتجات.

```
fig = plt.figure(figsize=(16,10))
ax1 = plt.subplot(211)
ax2 = plt.subplot(212, sharex = ax1)
reviews["asins"].value_counts().plot(kind="bar", ax=ax1,
title="ASIN Frequency")
np.log10(reviews["asins"].value_counts()).plot(kind="bar",
ax=ax2,
```




```
                         title="ASIN Frequency (Log10 Adjusted)")
plt.show()
```

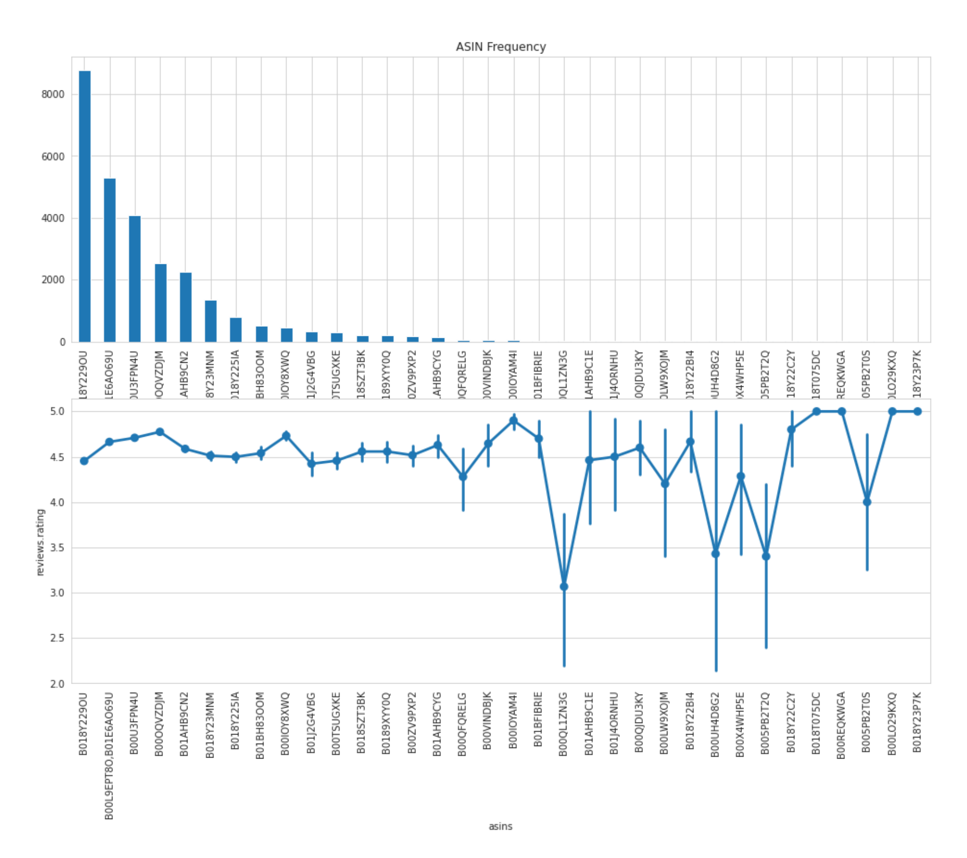




متوسط تصنيف مجموعة بيانات التدريب بالكامل

```
print(reviews["reviews.rating"].mean())

asins_count_ix = reviews["asins"].value_counts().index
plt.subplots(2,1,figsize=(16,12))
plt.subplot(2,1,1)
reviews["asins"].value_counts().plot(kind="bar", title="ASIN
Frequency")
plt.subplot(2,1,2)
sns.pointplot(x="asins", y="reviews.rating",
order=asins_count_ix, data=reviews)
plt.xticks(rotation=90)
plt.show()
```



تحليل المشاعر

باستخدام الميزات الموجودة، سننشئ مصنفاً يمكنه تحديد شعور المراجعة review's sentiment.

```
def sentiments(rating):
    if (rating == 5) or (rating == 4):
        return "Positive"
    elif rating == 3:
        return "Neutral"
    elif (rating == 2) or (rating == 1):
        return "Negative"
 #Add sentiments to the data
strat_train["Sentiment"] =
strat_train["reviews.rating"].apply(sentiments)
strat_test["Sentiment"] =
strat_test["reviews.rating"].apply(sentiments)
print(strat_train["Sentiment"][:20])
```

```
#Output-
4349          Positive
30776         Positive
28775         Neutral
```

```
1136       Positive
17803      Positive
7336       Positive
32638      Positive
13995      Positive
6728       Negative
22009      Positive
11047      Positive
22754      Positive
5578       Positive
11673      Positive
19168      Positive
14903      Positive
30843      Positive
5440       Positive
28940      Positive
31258      Positive
Name: Sentiment, dtype: object
```

المصدر:

https://thecleverprogrammer.com/2020/05/15/amazon-product-
/reviews-sentiment-analysis-with-machine-learning

21) تحليل المشاعر تجاه لقاح Covid-19 Covid-19 Vaccine Sentiment Analysis

غيّر وباء الفيروس التاجي (COVID-19) حياة الناس في جميع أنحاء العالم. لكن ظهور لقاحها أدى إلى ردود فعل إيجابية وسلبيةفي جميع أنحاء العالم.في هذه المقالة، سأقدم لك مشروع علم البيانات حول تحليل المشاعر تجاه لقاح Covid-19 باستخدام بايثون.

تحليل المشاعر تجاه لقاح Covid-19

قد لا تتوافق الرسائل الإعلامية دائمًا مع العلم لأن المعلومات المضللة والادعاءات التي لا أساس لها والشائعات يمكن أن تنتشر بسرعة. هذا ما رأيناه مع إدخال لقاح Covid-19.في مشروع علم البيانات هذا، نهدف إلى تحليل التغريدات المسجلة حول لقاح Covid-19 لتحليل مشاعر الناس تجاه اللقاح.

Twitter عبارة عن منصة تدوين مصغر وشبكات اجتماعية حيث ينشر المستخدمون رسائل تسمى "تغريدات tweets" ويتفاعلون معها. مع أكثر من 166 مليون مستخدم يوميًا، يعد Twitter مصدرًا قيّمًا للبيانات لأي مناقشة على وسائل التواصل الاجتماعي تتعلق بالأحداث الوطنية والعالمية. لذلك، تم جمع مجموعة البيانات الخاصة بمهمة تحليل المشاعر الخاصة بلقاح Covid-19 من Twitter.

مشروع علوم البيانات حول تحليل المشاعر المتعلقة بلقاح Covid-19

سأبدأ مهمة تحليل مشاعر لقاح Covid-19 عن طريق استيراد جميع مكتبات بايثون الضرورية:

مجموعة البيانات

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)
import re
import string
import nltk
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
import plotly.express as ex
import plotly.graph_objs as go
import plotly.offline as pyo
from plotly.subplots import make_subplots
pyo.init_notebook_mode()
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer as
SIA
from wordcloud import WordCloud,STOPWORDS
from pandas.plotting import autocorrelation_plot
from statsmodels.graphics.tsaplots import plot_acf
```

```
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.seasonal import seasonal_decompose
from nltk.util import ngrams
from nltk import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
import random
plt.rc('figure',figsize=(17,13))
```

في الكود أعلاه، سأقوم ببعض المعالجة المسبقة للنص لوظيفة مجموعة البيانات الخاصة بنا، والتي تحتوي على نص التغريدة. هدفنا هو إجراء تحليل المشاعر على بيانات نصية نظيفة لتجنب الضوضاء وأخطاء القراءة:

```
f_data = pd.read_csv('vaccination_tweets.csv')
f_data.text =f_data.text.str.lower()

#Remove twitter handlers
f_data.text = f_data.text.apply(lambda
x:re.sub('@[^\s]+','',x))

#remove hashtags
f_data.text = f_data.text.apply(lambda
x:re.sub(r'\B#\S+','',x))


# Remove URLS
f_data.text = f_data.text.apply(lambda x:re.sub(r"http\S+",
"", x))

# Remove all the special characters
f_data.text = f_data.text.apply(lambda x:'
'.join(re.findall(r'\w+', x)))

#remove all single characters
f_data.text = f_data.text.apply(lambda x:re.sub(r'\s+[a-zA-
Z]\s+', '', x))

# Substituting multiple spaces with single space
f_data.text = f_data.text.apply(lambda x:re.sub(r'\s+', ' ',
x, flags=re.I))
```

تحليل مشاعر Vander لقاح Covid-19

يعتمد تحليل المشاعر لـ VADER على قاموس يرسم الخصائص المعجمية إلى شدة عاطفية تسمى درجات المشاعر sentiment scores. يمكن الحصول على درجة المشاعر الخاصة بالنص من خلال تلخيص شدة intensity كل كلمةفي النص.

على سبيل المثال، ـ كلمات مثل "حب love" ، "نقدر appreciate" ، "سعيد happy" تنقل جميعها شعورًا إيجابيًا. أيضًا، يعد VADER ذكيًا بما يكفي لفهم السياق الأساسي لمثل هذه الكلمات، مثل "غير محبوب disliked" باعتباره بيانًا سلبيًا. ويتضمن أيضًا التركيز على الأحرف

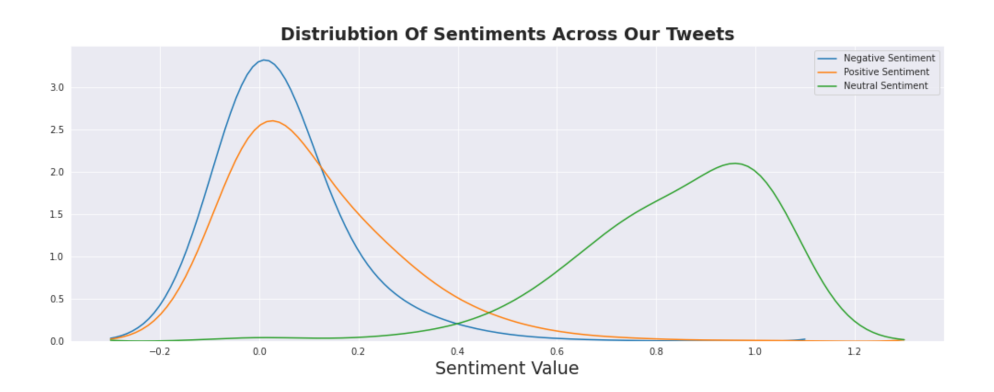
الكبيرة وعلامات الترقيم، مثل "استمتع ENJOY". الآن دعنا نجهز البيانات لتحليل VADER للمشاعر:

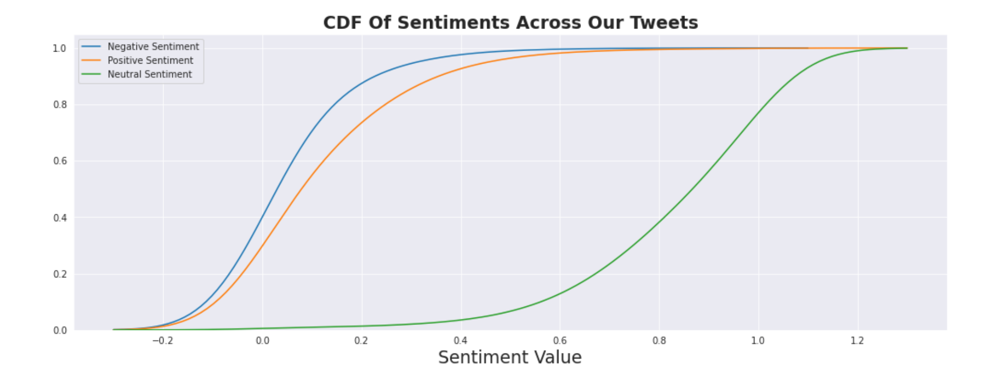
```
sid = SIA()
f_data['sentiments']            = f_data['text'].apply(lambda
x: sid.polarity_scores('
'.join(re.findall(r'\w+',x.lower()))))
f_data['Positive Sentiment']    =
f_data['sentiments'].apply(lambda x: x['pos']+1*(10**-6))
f_data['Neutral Sentiment']     =
f_data['sentiments'].apply(lambda x: x['neu']+1*(10**-6))
f_data['Negative Sentiment']    =
f_data['sentiments'].apply(lambda x: x['neg']+1*(10**-6))

f_data.drop(columns=['sentiments'],inplace=True)
```

تحليل البيانات الاستكشافية

```
plt.subplot(2,1,1)
plt.title('Distriubtion Of Sentiments Across Our
Tweets',fontsize=19,fontweight='bold')
sns.kdeplot(f_data['Negative Sentiment'],bw=0.1)
sns.kdeplot(f_data['Positive Sentiment'],bw=0.1)
sns.kdeplot(f_data['Neutral Sentiment'],bw=0.1)
plt.xlabel('Sentiment Value',fontsize=19)
plt.subplot(2,1,2)
plt.title('CDF Of Sentiments Across Our
Tweets',fontsize=19,fontweight='bold')
sns.kdeplot(f_data['Negative
Sentiment'],bw=0.1,cumulative=True)
sns.kdeplot(f_data['Positive
Sentiment'],bw=0.1,cumulative=True)
sns.kdeplot(f_data['Neutral
Sentiment'],bw=0.1,cumulative=True)
plt.xlabel('Sentiment Value',fontsize=19)
plt.show()
```





يمكنك ملاحظة أن توزيعات المشاعر تتبع التوزيع الطبيعي normal distribution؛ المشاعر السلبية والإيجابية متشابهة جداً، مما يوحي بعدم وجود فروق ذات دلالة إحصائيةفي قوة المشاعر الإيجابية والسلبيةفي بياناتنا.

ومن الواضح أيضًا أن الشعور السائد محايد؛ من الغريب أن معظم التغريدات لا تبدو أشبه بمشاعر إيجابية أو سلبية محايدة.

تحليل المشاعر باستخدام بايثون

لنبدأ الآن بتحليل الحد الأقصى للمشاعر السلبية والأكثر إيجابية:

```
#Sorting And Feature Engineering
f_data = f_data.sort_values(by='date')
ft_data=f_data.copy()
ft_data['date'] = pd.to_datetime(f_data['date']).dt.date

ft_data['year']         =
pd.DatetimeIndex(ft_data['date']).year
ft_data['month']        =
pd.DatetimeIndex(ft_data['date']).month
ft_data['day']          =
pd.DatetimeIndex(ft_data['date']).day
ft_data['day_of_year']  =
pd.DatetimeIndex(ft_data['date']).dayofyear
ft_data['quarter']      =
pd.DatetimeIndex(ft_data['date']).quarter
ft_data['season']       = ft_data.month%12 // 3 + 1

plt.subplot(2,1,1)
plt.title('Selecting A Cut-Off For Most Positive/Negative
Tweets',fontsize=19,fontweight='bold')

ax0 = sns.kdeplot(f_data['Negative Sentiment'],bw=0.1)

kde_x, kde_y = ax0.lines[0].get_data()
ax0.fill_between(kde_x, kde_y, where=(kde_x>0.25) ,
                interpolate=True, color='b')
```

```
plt.annotate('Cut-Off For Most Negative Tweets', xy=(0.25,
0.5), xytext=(0.4, 2),
            arrowprops=dict(facecolor='red',
shrink=0.05),fontsize=16,fontweight='bold')

ax0.axvline(f_data['Negative Sentiment'].mean(), color='r',
linestyle='--')
ax0.axvline(f_data['Negative Sentiment'].median(),
color='tab:orange', linestyle='-')
plt.legend({'PDF':f_data['Negative Sentiment'],r'Mean:
{:.2f}'.format(f_data['Negative
Sentiment'].mean()):f_data['Negative Sentiment'].mean(),
            r'Median: {:.2f}'.format(f_data['Negative
Sentiment'].median()):f_data['Negative Sentiment'].median()})

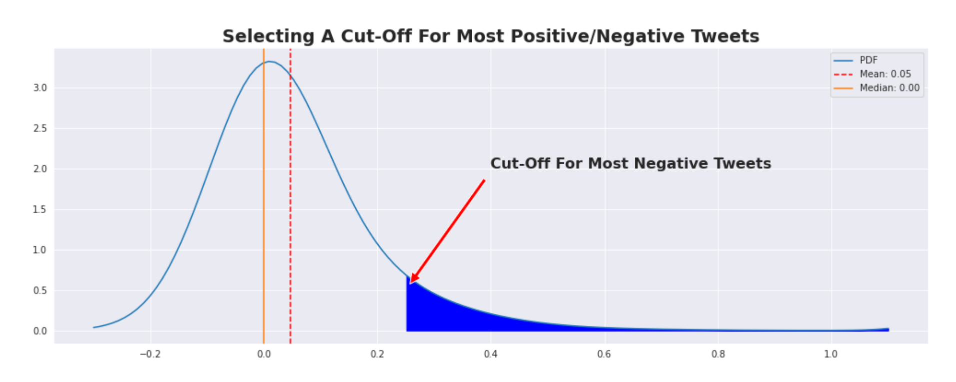
plt.subplot(2,1,2)

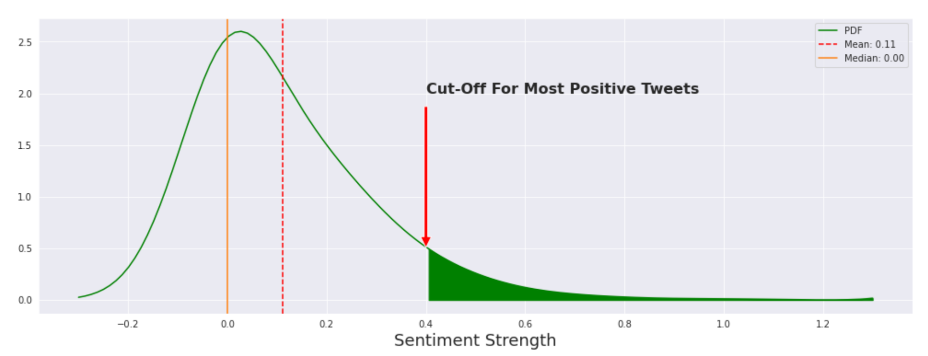
ax1 = sns.kdeplot(f_data['Positive
Sentiment'],bw=0.1,color='green')

plt.annotate('Cut-Off For Most Positive Tweets', xy=(0.4,
0.43), xytext=(0.4, 2),
            arrowprops=dict(facecolor='red',
shrink=0.05),fontsize=16,fontweight='bold')
kde_x, kde_y = ax1.lines[0].get_data()
ax1.fill_between(kde_x, kde_y, where=(kde_x>0.4) ,
                interpolate=True, color='green')
ax1.set_xlabel('Sentiment Strength',fontsize=18)


ax1.axvline(f_data['Positive Sentiment'].mean(), color='r',
linestyle='--')
ax1.axvline(f_data['Positive Sentiment'].median(),
color='tab:orange', linestyle='-')
plt.legend({'PDF':f_data['Positive Sentiment'],r'Mean:
{:.2f}'.format(f_data['Positive
Sentiment'].mean()):f_data['Positive Sentiment'].mean(),
            r'Median: {:.2f}'.format(f_data['Positive
Sentiment'].median()):f_data['Positive Sentiment'].median()})

plt.show()
```





الآن دعونا نتخيل أكثر المشاعر سلبية وإيجابية:

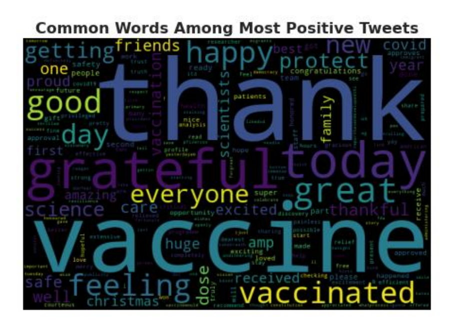
```
Most_Positive = f_data[f_data['Positive
Sentiment'].between(0.4,1)]
Most_Negative = f_data[f_data['Negative
Sentiment'].between(0.25,1)]

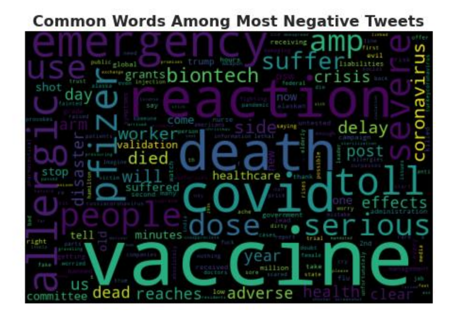
Most_Positive_text = ' '.join(Most_Positive.text)
Most_Negative_text = ' '.join(Most_Negative.text)


pwc = WordCloud(width=600,height=400,collocations =
False).generate(Most_Positive_text)
nwc = WordCloud(width=600,height=400,collocations =
False).generate(Most_Negative_text)

plt.subplot(1,2,1)
plt.title('Common Words Among Most Positive
Tweets',fontsize=16,fontweight='bold')
plt.imshow(pwc)
plt.axis('off')
plt.subplot(1,2,2)
plt.title('Common Words Among Most Negative
Tweets',fontsize=16,fontweight='bold')
plt.imshow(nwc)
plt.axis('off')

plt.show()
```





أخيرًا، دعنا نلقي نظرة على أهم 10 مشاعر سلبية وإيجابية:

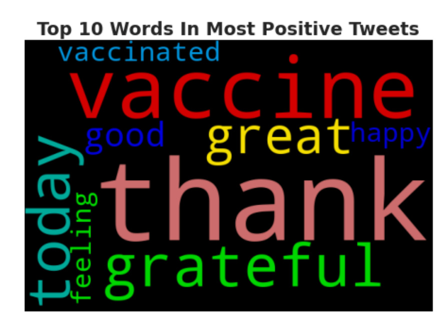
```
l_t = Most_Positive_text

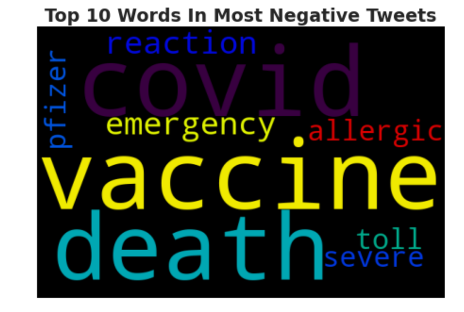
w1_dict = dict()
for word in l_t.split():
    w= word.strip()
    if w in STOPWORDS:
        continue
    else:
        w1_dict[w] = w1_dict.get(w,0)+1
w1_dict = {k: v for k, v in sorted(w1_dict.items(), key=lambda
item: item[1],reverse=True)}

l_t = Most_Negative_text
w2_dict = dict()
for word in l_t.split():
    w= word.strip()
    if w in STOPWORDS:
        continue
    else:
        w2_dict[w] = w2_dict.get(w,0)+1
w2_dict = {k: v for k, v in sorted(w2_dict.items(), key=lambda
item: item[1],reverse=True)}

top_10_pos = list(w1_dict.keys())[:10]
top_10_neg = list(w2_dict.keys())[:10]

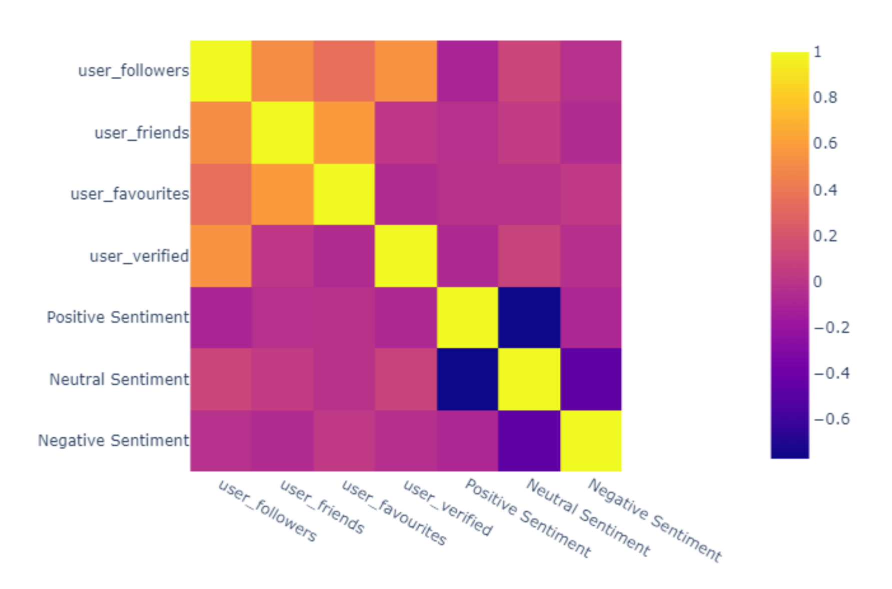
plt.subplot(1,2,1)
w_c = WordCloud(width=600,height=400,collocations =
False,colormap='nipy_spectral').generate(' '.join(top_10_pos))
plt.title('Top 10 Words In Most Positive
Tweets',fontsize=19,fontweight='bold')
plt.imshow(w_c)
plt.axis('off')
plt.subplot(1,2,2)
w_c = WordCloud(width=600,height=400,collocations =
False,colormap='nipy_spectral').generate(' '.join(top_10_neg))
plt.title('Top 10 Words In Most Negative
Tweets',fontsize=19,fontweight='bold')
plt.imshow(w_c)
plt.axis('off')
plt.show()
```





دعنا الآن نلقي نظرة على العلاقة بين التغريدات والميزات الرقمية الأخرىفي مجموعة البيانات:

```
ex.imshow(f_data[['user_followers','user_friends','user_favour
ites','user_verified','Positive Sentiment,'
                      'Neutral Sentiment','Negative
Sentiment']].corr('spearman'),title='Spearman Correlation')
```



لسوء الحظ، لا نرى أي ارتباط كبير بين الشعور بالتغريدة وأي خاصية رقمية أخرى موجودةفي مجموعة البيانات الخاصة بنا، خاصة تلك التي تصف المستخدمين. آمل أن تكون قد أحببت هذه المقالة حول مشروع علم البيانات حول تحليل لقاح Covid-19 باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/01/11/covid-19-vaccine-sentiment-analysis/

22) تحليل مشاعر منشورات فيسبوك Facebook Posts Sentiment Analysis

يعد Facebook منصة جيدة جداً لأداء مهمة تحليل المشاعر لأن المستخدمين أحرارفي التعبير عن آرائهم حول أي موضوع سواء كان سياسيًا أو بيئيًا، وللمستخدمين حرية مشاركة آرائهم.في هذه المقالة، سأقدم لك تحليل المشاعرفي منشورات Facebook باستخدام التعلم الآلي باستخدام بايثون.

تحليل مشاعر Facebook منشورات باستخدام التعلم الآلي

بالنسبة لمهمة تحليل المشاعر الخاصة بمنشورات Facebook، فأنت بحاجة إلى استخراج بياناتك من Facebook أولاً، وهي مهمة سهلة للغاية، ما عليك سوى اتباع الخطوات المذكورة أدناه:

1. انتقل إلى الإعدادات والخصوصية.
2. ثم انتقل إلى الإعدادات.
3. من اليسار، انقر فوق معلومات Facebook الخاصة بك.
4. انقر فوق عرضفي تنزيل المعلومات الخاصة بك.
5. ثم حدد المنشورات فقط وانقر فوق إنشاء ملف.

سيرسل لك Facebook إشعارًافي الـ 60 دقيقة القادمة لتنزيل بياناتك. يجب عليك البحث عن ملف "your_posts_1.json"في البيانات التي تم تنزيلها ، لأننا نحتاج فقط إلى هذه البيانات لمهمة تحليل مشاعر منشورات Facebook باستخدام بايثون.

تحليل مشاعر Facebook منشورات باستخدام بايثون

لنبدأ الآن بمهمة تحليل المشاعرفي منشورات Facebook باستخدام التعلم الآلي باستخدام بايثون. سأبدأ باستيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.stem import LancasterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from nltk.probability import FreqDist
import re
import unicodedata
import nltk
import json
import inflect
import matplotlib.pyplot as plt
```

```
import json
# load json into python, assign to 'data'
with open('your_posts_1.json') as file:
    data = json.load(file)
print(type(data))      # a list
print(type(data[0]))  # first object in the list: a dictionary
print(len(data))
```

```
<class 'list'>
<class 'dict'>
5038
```

نحتاج الآن إلى استخراج الأوصاف من المنشورات حتى نتمكن من تحليل مشاعر المنشورات. سأخزن البيانات النصية لمنشوراتنافي قائمة:

```
# create empty list
empty_lst = []
# multiple nested loops to store all post in empty list
for dct in data:
    for k, v in dct.items():
        if k == 'data':
            if len(v) > 0:
                for k_i, v_i in v[0].items():
                    if k_i == 'post':
                        empty_lst.append(v_i)
print("This is the empty list: ", empty_lst)
print("\nLength of list: ", len(empty_lst))
for i in empty_lst:
  print(i)
```

```
Length of list:  2885
Krish Naik created a Video for með Must watch!
Amazing Work by @[100010286773956:2048:Hritika Aggarwal]ðº
.......
```

كما ترى من الإخراج أعلاه، لدينا قائمة تحتوي على النص الذي كتبتهفي منشوراتي على Facebook. لقد نشرت فقط حول علم البيانات والتعلم الآلي على Facebook، لذا فإن هذه البيانات ليست مثيرة للاهتمام، ولكن يمكنك تنفيذ نفس الكود إذا كنت قد نشرت آراء سياسية. الكود الذي كتبتهفي شكل يمكن تنفيذه على أي نوع بيانات.

تحضير البيانات

سأقوم الآن بإجراء عملية الترميز tokenization على البيانات النصية التي قمنا بتخزينها للتوفي قائمة:

```
nltk.download('punkt')
nested_sent_token = [nltk.sent_tokenize(lst) for lst in
empty_lst]
```

```
# flatten list, len: 3241
flat_sent_token = [item for sublist in nested_sent_token for
item in sublist]
print("Flatten sentence token: ", len(flat_sent_token))
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
Flatten sentence token:  3923
```

نحتاج الآن إلى تسوية normalize البيانات، هنا سأقوم بإنشاء بعض الدوال من أجل:

- حذف أحرف غير ASCII.
- كل الكلمات بحروف صغيرة.
- حذف علامات الترقيم.
- استبدال الرقم.
- كلمات التوقف.

دعنا نرى كيفية إنشاء دوال لأداء جميع المهام المذكورة أعلاه لإعداد البيانات لتحليل مشاعر منشورات Facebook:

```
def remove_non_ascii(words):
    new_words = []
    for word in words:
        new_word = unicodedata.normalize('NFKD', word).encode(
            'ascii', 'ignore').decode('utf-8', 'ignore')
        new_words.append(new_word)
    return new_words

# To LowerCase
def to_lowercase(words):
    new_words = []
    for word in words:
        new_word = word.lower()
        new_words.append(new_word)
    return new_words

# Remove Punctuation
def remove_punctuation(words):
    new_words = []
    for word in words:
        new_word = re.sub(r'[^\w\s]', '', word)
        if new_word != '':
            new_words.append(new_word)
    return new_words

# Replace Numbers with Textual Representations
def replace_numbers(words):
    p = inflect.engine()
```

```
    new_words = []
    for word in words:
        if word.isdigit():
            new_word = p.number_to_words(word)
            new_words.append(new_word)
        else:
            new_words.append(word)
    return new_words

# Remove Stopwords
def remove_stopwords(words):
    new_words = []
    for word in words:
        if word not in stopwords.words('english'):
            new_words.append(word)
    return new_words

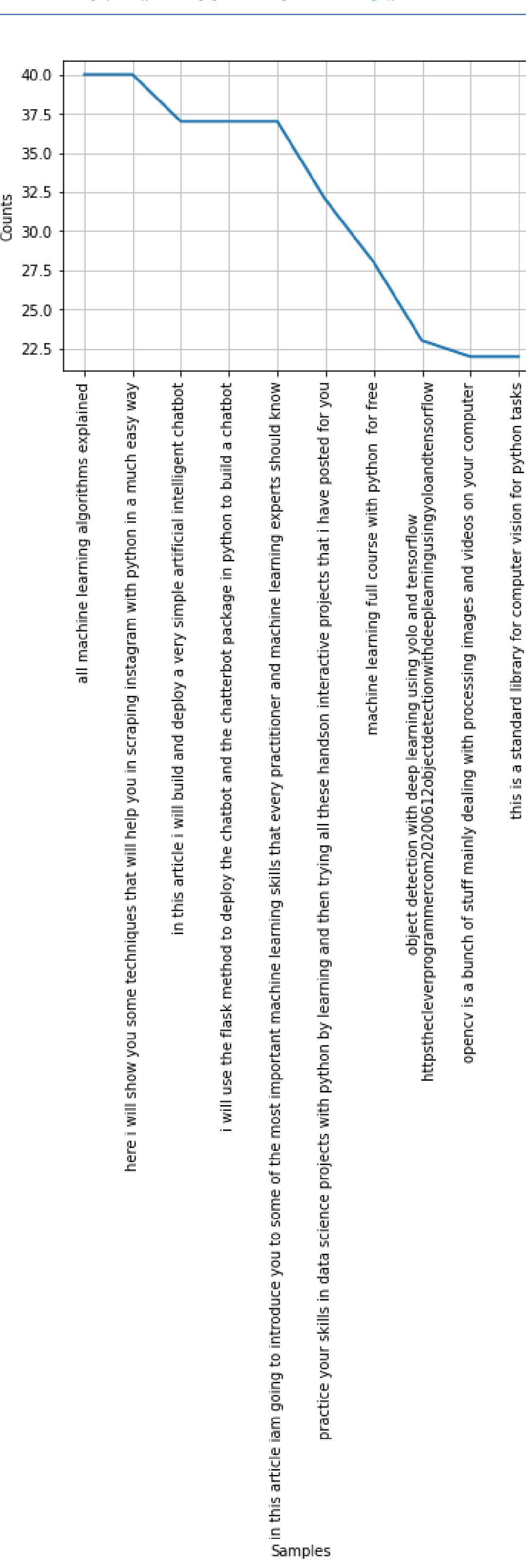
# Combine all functions into Normalize() function
def normalize(words):
    words = remove_non_ascii(words)
    words = to_lowercase(words)
    words = remove_punctuation(words)
    words = replace_numbers(words)
    words = remove_stopwords(words)
    return words

nltk.download('stopwords')
sents = normalize(flat_sent_token)
print("Length of sentences list: ", len(sents))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
Length of sentences list:  3866
```

الآن دعونا نرى ما يمكننا تصويره في هذه البيانات. سوف ألقي نظرة على الجمل الأكثر شيوعًا لمعرفة أكثر ما أشاركه في منشوراتي على Facebook:

```
from nltk.probability import FreqDist
# Find frequency of sentence
fdist_sent = FreqDist(sents)
fdist_sent.most_common(10)
# Plot
fdist_sent.plot(10)
```


Counts
40.0
37.5
35.0
32.5
30.0
27.5
25.0
22.5
all machine learning algorithms explained
here i will show you some techniques that will help you in scraping instagram with python in a much easy way
in this article i will build and deploy a very simple artificial intelligent chatbot
i will use the flask method to deploy the chatbot and the chatterbot package in python to build a chatbot
in this article iam going to introduce you to some of the most important machine learning skills that every practitioner and machine learning experts should know
practice your skills in data science projects with python by learning and then trying all these handson interactive projects that i have posted for you
machine learning full course with python for free
object detection with deep learning using yolo and tensorflow
httpsthecleverprogrammercom20200612objectdetectionwithdeeplearningusingyoloandtensorflow
opencv is a bunch of stuff mainly dealing with processing images and videos on your computer
this is a standard library for computer vision for python tasks
Samples

الخطوة النهائية: تحليل مشاعر منشورات Facebook

الآن الخطوة الأخيرة هي تحليل مشاعر النص المكتوب في وصف المنشور وإنشاء تقرير لتحليل المشاعر على Facebook يوضح ما إذا كانت المشاعر محايدة neutral أو إيجابية positive أو سلبية negative:

```
nltk.download('vader_lexicon')
sid = SentimentIntensityAnalyzer()
sentiment = []
sentiment2 = []
for sent in sents:
    sent1 = sent
    sent_scores = sid.polarity_scores(sent1)
    for x, y in sent_scores.items():
        sentiment2.append((x, y))
    sentiment.append((sent1, sent_scores))
    # print(sentiment)
# sentiment
cols = ['sentence', 'numbers']
result = pd.DataFrame(sentiment, columns=cols)
print("First five rows of results: ", result.head())
# sentiment2
cols2 = ['label', 'values']
result2 = pd.DataFrame(sentiment2, columns=cols2)
print("First five rows of results2: ", result2.head())
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
First five rows of results:                                    sentence
numbers
0       krish naik created a video for me must watch  {'neg': 0.0, 'neu': 0.778,
'pos': 0.222, 'comp...
1  amazing work by 1000102867739562048hritika agg...  {'neg': 0.0, 'neu': 0.513,
'pos': 0.487, 'comp...
2  in this article iall walk you through how to c...  {'neg': 0.0, 'neu': 0.833,
'pos': 0.167, 'comp...
3  machine learning full course with python  for ...  {'neg': 0.0, 'neu': 0.68,
'pos': 0.32, 'compou...
4  practice your skills in data science projects ...  {'neg': 0.0, 'neu': 1.0, 'pos':
0.0, 'compound...
First five rows of results2:        label  values
0       neg   0.000
1       neu   0.778
2       pos   0.222
3  compound   0.250
4       neg   0.000
```

آمل أن تكون قد أحببت هذه المقالة على تحليل مشاعر منشورات Facebook باستخدام التعلم الآلي باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/01/31/facebook-posts-sentiment-analysis/

23) تحليل تصنيف الفيلم باستخدام بايثون Movie Rating Analysis using Python

نشاهد جميعًا الأفلام للترفيه، والبعض منا لا يقيمها أبدًا، بينما يقوم بعض المشاهدين دائمًا بتقييم كل فيلم يشاهدونه. يساعد هذا النوع من المشاهدين في تصنيف الأفلام للأشخاص الذين يراجعون مراجعات الفيلم قبل مشاهدة أي فيلم للتأكد من أنهم على وشك مشاهدة فيلم جيد. لذلك، إذا كنت جديدًا في علم البيانات وترغب في معرفة كيفية تحليل تقييمات الأفلام باستخدام لغة برمجة بايثون، فهذه المقالة مناسبة لك. في هذه المقالة، سوف أطلعك على مهمة تحليل تصنيف الأفلام Movie Rating Analysis باستخدام بايثون.

تحليل تصنيف الفيلم باستخدام بايثون

يساعد تحليل التصنيف الذي قدمه مشاهدو الفيلم العديد من الأشخاص في تحديد ما إذا كانوا سيشاهدون هذا الفيلم أم لا. لذلك، بالنسبة لمهمة تحليل تصنيف الفيلم، تحتاج أولاً إلى مجموعة بيانات تحتوي على بيانات حول التصنيفات التي قدمها كل مشاهد. لهذه المهمة، قمت بجمع مجموعة بيانات من Kaggle تحتوي على ملفين:

1. يحتوي ملف واحد على بيانات حول معرف الفيلم وعنوانه ونوعه.
2. ويحتوي الملف الآخر على معرف المستخدم ومعرف الفيلم والتقييمات التي قدمها المستخدم والطابع الزمني timestamp للتصنيفات.

يمكنك تنزيل مجموعتي البيانات هاتين من هنا.

لنبدأ الآن بمهمة تحليل تصنيف الأفلام عن طريق استيراد مكتبات بايثون ومجموعات البيانات الضرورية:

```
import numpy as np
import pandas as pd
movies = pd.read_csv("movies.dat", delimiter='::')
print(movies.head())
```

```
0      10                      La sortie des usines Lumière (1895)     Documentary|Short
1      12                           The Arrival of a Train (1896)     Documentary|Short
2      25  The Oxford and Cambridge University Boat Race ...                   NaN
3      91                            Le manoir du diable (1896)          Short|Horror
4     131                             Une nuit terrible (1896)   Short|Comedy|Horror
```

في الكود أعلاه، قمت فقط باستيراد مجموعة بيانات الأفلام التي لا تحتوي على أي أسماء أعمدة، لذلك دعونا نحدد أسماء الأعمدة:

```
movies.columns = ["ID", "Title", "Genre"]
```

```
print(movies.head()
```

```
    ID                                              Title                 Genre
0   10                 La sortie des usines Lumière (1895)    Documentary|Short
1   12                      The Arrival of a Train (1896)    Documentary|Short
2   25  The Oxford and Cambridge University Boat Race ...                  NaN
3   91                         Le manoir du diable (1896)         Short|Horror
4  131                          Une nuit terrible (1896)  Short|Comedy|Horror
```

الآن دعنا نستورد مجموعة بيانات التصنيفات ratings:

```
ratings = pd.read_csv("ratings.dat", delimiter='::')
print(ratings.head())
```

```
   1  0114508  8  1381006850
0  2   499549  9  1376753198
1  2  1305591  8  1376742507
2  2  1428538  1  1371307089
3  3    75314  1  1595468524
4  3   102926  9  1590148016
```

لا تحتوي مجموعة بيانات التصنيف أيضًا على أي أسماء أعمدة، لذلك دعونا نحدد أسماء الأعمدة لهذه البيانات أيضًا:

```
ratings.columns = ["User", "ID", "Ratings", "Timestamp"]
print(ratings.head())
```

```
   User       ID  Ratings   Timestamp
0     2   499549        9  1376753198
1     2  1305591        8  1376742507
2     2  1428538        1  1371307089
3     3    75314        1  1595468524
4     3   102926        9  1590148016
```

سأقوم الآن بدمج مجموعتي البيانات هاتينفي واحدة، تحتوي مجموعتي البيانات هاتين على عمود مشترك كمعرف (ID)، والذي يحتوي على معرف الفيلم، لذلك يمكننا استخدام هذا العمود كعمود مشترك لدمج مجموعتي البيانات:

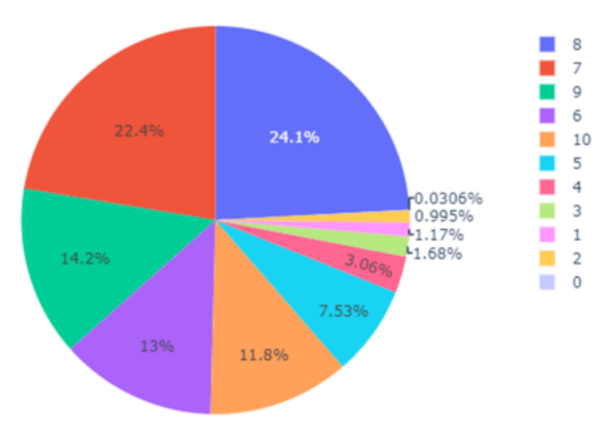
```
data = pd.merge(movies, ratings, on=["ID", "ID"])
print(data.head())
```

```
   ID                                              Title  ... Ratings   Timestamp
0  10                La sortie des usines Lumière (1895)  ...      10  1412878553
1  12                      The Arrival of a Train (1896)  ...      10  1439248579
2  25  The Oxford and Cambridge University Boat Race ...  ...       8  1488189899
3  91                         Le manoir du diable (1896)  ...       6  1385233195
4  91                         Le manoir du diable (1896)  ...       5  1532347349

[5 rows x 6 columns]
```

نظرًا لأنها مهمة على مستوى المبتدئين، سأقوم أولاً بإلقاء نظرة على توزيع تصنيفات جميع الأفلام التي قدمها المشاهدون:

```
ratings = data["Ratings"].value_counts()
numbers = ratings.index
quantity = ratings.values
import plotly.express as px
fig = px.pie(data, values=quantity, names=numbers)
fig.show()
```



لذلك، وفقًا للمخطط الدائري أعلاه، تم تصنيف معظم الأفلام 8 من قبل المستخدمين. من الشكل أعلاه، يمكن القول إن معظم الأفلام تم تصنيفها بشكل إيجابي.

نظرًا لأن الرقم 10 هو أعلى تصنيف يمكن للمشاهد تقديمه، فلنلقِ نظرة على أفضل 10 أفلام حصلت على 10 تقييمات من قبل المشاهدين:

```
data2 = data.query("Ratings == 10")
print(data2["Title"].value_counts().head(10))
```

```
Joker (2019)                     1479
Interstellar (2014)              1382
1917 (2019)                       819
Avengers: Endgame (2019)          808
The Shawshank Redemption (1994)   699
Gravity (2013)                    653
The Wolf of Wall Street (2013)    581
Hacksaw Ridge (2016)              570
Avengers: Infinity War (2018)     534
La La Land (2016)                 510
Name: Title, dtype: int64
```

لذلك، وفقًا لمجموعة البيانات هذه، حصل Joker (2019) على أعلى عدد 10 تقييمات من المشاهدين. هذه هي الطريقة التي يمكنك بها تحليل تقييمات الأفلام باستخدام بايثون كمبتدئ في علم البيانات.

الملخص

هذه هي الطريقة التي يمكنك بها إجراء تحليل تصنيف الفيلم باستخدام لغة برمجة بايثون كمبتدئ في علم البيانات. يساعد تحليل التصنيفات التي قدمها مشاهدو الفيلم العديد من الأشخاص في تحديد ما إذا كانوا سيشاهدون هذا الفيلم أم لا. آمل أن تكون قد أحببت هذه المقالة حول تحليل تصنيف الأفلام باستخدام بايثون.

المصدر:

https://thecleverprogrammer.com/2021/09/22/movie-rating-analysis-/using-python

24) تحليل المشاعر على تويتر باستخدام التعلم الالي Twitter Sentiment Analysis using Machine Learning

Twitter هو أحد منصات التواصل الاجتماعي حيث يتمتع الأشخاص بحرية مشاركة آرائهم حول أي موضوع. نرى أحيانًا مناقشة قوية على Twitter حول رأي شخص ما تؤدي أحيانًا إلى مجموعة من التغريدات السلبية. مع وضع ذلك في الاعتبار، إذا كنت تريد معرفة كيفية إجراء تحليل المشاعر على Twitter، فهذه المقالة مناسبة لك. في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعر على Twitter باستخدام بايثون.

تحليل المشاعر على تويتر

تحليل المشاعر Sentiment analysis مهمة معالجة اللغة الطبيعية. يجب على جميع منصات وسائل التواصل الاجتماعي مراقبة مشاعر المشاركين في المناقشة. نرى في الغالب آراء سلبية على تويتر عندما تكون المناقشة سياسية. لذلك، يجب أن تستمر كل منصة في تحليل المشاعر للعثور على نوع الأشخاص الذين ينشرون الكراهية والسلبية على نظامهم الأساسي.

بالنسبة لمهمة تحليل المشاعر على Twitter، قمت بجمع مجموعة بيانات من Kaggle تحتوي على تغريدات حول مناقشة طويلة داخل مجموعة من المستخدمين. مهمتنا هنا هي تحديد عدد التغريدات السلبية والإيجابية حتى نتمكن من إعطاء نتيجة. لذلك، في القسم أدناه، سأقدم لك مهمة تحليل المشاعر على Twitter باستخدام بايثون.

تحليل المشاعر على Twitter باستخدام بايثون

لنبدأ مهمة تحليل المشاعر على Twitter من خلال استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import re
import nltk
import nltk

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-
data/master/twitter.csv")
print(data.head())
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                               tweet
0  !!! RT @mayasolovely: As a woman you shouldn't...
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
```

يحتوي عمود التغريدات tweet column في مجموعة البيانات أعلاه على التغريدات التي نحتاج إلى استخدامها لتحليل مشاعر المشاركين في المناقشة. ولكن للمضي قدمًا، يتعين علينا تنظيف الكثير من الأخطاء والرموز الخاصة الأخرى لأن هذه التغريدات تحتوي على الكثير من الأخطاء اللغوية. إذن إليك كيف يمكننا تنظيف عمود التغريدة:

```
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["tweet"] = data["tweet"].apply(clean)
```

الآن، الخطوة التالية هي حساب درجات المشاعر لهذه التغريدات وتعيين تسمية للتغريدات على أنها إيجابية positive أو سلبية negative أو محايدة neutral . إليك كيفية حساب درجات المشاعر في التغريدات:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i in
data["tweet"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i in
data["tweet"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i in
data["tweet"]]
```

الآن سأختار فقط الأعمدة من هذه البيانات التي نحتاجها لبقية مهمة تحليل المشاعر على Twitter:

```
data = data[["tweet", "Positive",
             "Negative", "Neutral"]]
print(data.head())
```

```
                                         tweet  Positive  Negative  \
0  rt mayasolov woman shouldnt complain clean ho...     0.147     0.157
1  rt  boy dat coldtyga dwn bad cuffin dat hoe  ...     0.000     0.280
2  rt urkindofbrand dawg rt  ever fuck bitch sta...     0.000     0.577
3            rt cganderson vivabas look like tranni     0.333     0.000
4  rt shenikarobert shit hear might true might f...     0.154     0.407

   Neutral
0    0.696
1    0.720
2    0.423
3    0.667
4    0.440
```

دعنا الآن نلقي نظرة على التصنيف الأكثر شيوعًا المخصص للتغريدات وفقًا لدرجات المشاعر:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

Neutral 🙂

لذا فإن معظم التغريدات محايدة، ما يعني أنها ليست إيجابية ولا سلبية. الآن دعنا نلقي نظرة على إجمالي درجات المشاعر:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

Positive: 2880.086000000009
Negative: 7201.020999999922
Neutral: 14696.887999999733

مجموع التغريدات المحايدة أعلى بكثير من السلبية والإيجابية، لكن من بين جميع التغريدات السلبية أكبر من التغريدات الإيجابية، لذلك يمكننا القول إن معظم الآراء سلبية.

الملخص

هذه هي الطريقة التي يمكنك بها أداء مهمة تحليل المشاعر على Twitter باستخدام لغة برمجة بايثون. تحليل المشاعر مهمة معالجة اللغة الطبيعية. تحتاج جميع منصات وسائل التواصل الاجتماعي إلى التحقق من مشاعر الأشخاص المشاركين في المناقشة. آمل أن تكون قد أحببت هذا المقال على تحليل المشاعر على Twitter باستخدام بايثون.